Advanced Review

In silico toxicology: comprehensive benchmarking of multi-label classification methods applied to chemical toxicity data

Arwa B. Raies and Vladimir B. Bajic*

One goal of toxicity testing, among others, is identifying harmful effects of chemicals. Given the high demand for toxicity tests, it is necessary to conduct these tests for multiple toxicity endpoints for the same compound. Current computational toxicology methods aim at developing models mainly to predict a single toxicity endpoint. When chemicals cause several toxicity effects, one model is generated to predict toxicity for each endpoint, which can be labor and computationally intensive when the number of toxicity endpoints is large. Additionally, this approach does not take into consideration possible correlation between the endpoints. Therefore, there has been a recent shift in computational toxicity studies toward generating predictive models able to predict several toxicity endpoints by utilizing correlations between these endpoints. Applying such correlations jointly with compounds' features may improve model's performance and reduce the number of required models. This can be achieved through multi-label classification methods. These methods have not undergone comprehensive benchmarking in the domain of predictive toxicology. Therefore, we performed extensive benchmarking and analysis of over 19,000 multi-label classification models generated using combinations of the state-of-the-art methods. The methods have been evaluated from different perspectives using various metrics to assess their effectiveness. We were able to illustrate variability in the performance of the methods under several conditions. This review will help researchers to select the most suitable method for the problem at hand and provide a baseline for evaluating new approaches. Based on this analysis, we provided recommendations for potential future directions in this area.

How to cite this article:
WIREs Comput Mol Sci 2018, 8:e1352. doi: 10.1002/wcms.1352

INTRODUCTION

Toxicity testing aims at identifying harmful effects of chemicals on humans, animals, plants, and the environment. Additionally, there are many benefits of toxicity testing[1]: to determine safety of compounds and identify safe doses; to categorize potential toxic effects through different routes of exposure (e.g., oral, dermal, or inhalation) and frequency of exposure; to determine mode and mechanism of action; to explain observations of effects in different groups of the population (e.g., based on gender or

Additional Supporting Information may be found in the online version of this article.

*Correspondence to: vladimir.bajic@kaust.edu.sa

Computational Bioscience Research Centre (CBRC), Computer, Electrical and Mathematical Sciences and Engineering Division (CEMSE), King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia

Conflict of interest: The authors have declared no conflicts of interest for this article.

age); and to verify *in vitro* or *in silico* testing results.[1] It is a necessary step in many processes including drugs design[2]; identification of environmental hazards, such as chemical pollutants[3]; manufacturing chemical products, such as pesticides[3]; and synthesis of food products, such as food additives.[4] Given the high demand for toxicity tests, it is crucial to conduct these tests for multiple toxicity endpoints for the same compound. For a long time, *in vivo* testing (i.e., using animal models or cell lines) has been the primary toxicity assessment method.[2] Following the development of High Throughput Screening, *in vitro* toxicity testing in a large number of cultured cells or cell lines became feasible.[5] In contrast, *in silico* toxicology relies on using computational resources for toxicity assessment.[6] *In silico* toxicology has contributed considerably to toxicity testing and its first developments occurred as early as 1934.[7] During the last century, *in silico* toxicology has evolved significantly to provide various computational tools for gathering, modeling, simulating, and visualizing toxicity data.[6]

Several methods have been developed to generate toxicity prediction models.[6] Current *in silico* toxicology methods aim at developing models mainly to predict a single toxicity endpoint.[6] When chemicals cause several toxicity effects, one model is generated for each endpoint[8] (i.e., the binary relevance method[9]), which can be laborious and computationally intensive when the number of toxicity endpoints is large. It should be noted that there are some systems (e.g., Derek Nexus,[10] ToxTree,[11,12] and OECD QSAR toolbox[13]) that include many pre-built models to predict different toxicity endpoints.[6] These tools contain many pre-built models each aimed at predicting a single endpoint. While it is easy to use the existing pre-built models for each endpoint, it is labor intensive to develop these models in the first place. With advancements in High Throughput Screening, it became feasible to measure hundreds of toxicity endpoints per compound.[5] Developing one model for each newly measured endpoint would require developing hundreds of models. This is inefficient and impractical. Additionally, this approach does not take into consideration possible correlation between the endpoints, and it relies entirely on features of compounds. However, it has been found that some toxicity endpoints may correlate.[8,14] In 2006, analysis of data from the US Food and Drug Administration showed that there are high correlations between four composite toxicity categories (gene mutation, DNA damage, clastogenicity, and reproductive toxicity) with carcinogenicity in rodents.[14]

Therefore, there was a recent shift in computational toxicity toward generating predictive models able to predict several toxicity endpoints by utilizing correlations between these endpoints. This can be achieved through multi-label classification methods. Applying such associations jointly with compounds' features may improve model performance and reduce the number of generated models. Developments have been made to create multi-label classification models using toxicity data sets such as the ToxCast data set,[15,16] Tox21 data sets,[17–19] Accelrys Toxicity Database,[20,21] and RepDose and ELINC data sets.[22,23] However, a recurrent problem in these studies, caused by missing labels in toxicity data sets, illustrated a major challenge in applying multi-label classification approaches to toxicity prediction. Toxicity profiles of some compounds are unknown across all toxicity phenotypes, either because such data are unavailable (e.g., compounds are not tested for all toxicity phenotypes), or it may be hard to find (e.g., being scattered in scientific literature). Determining relationships between endpoints is more difficult if toxicity data are not available for all toxicity endpoints in a data set. In such cases, some studies excluded compounds whose toxicity is unknown across all toxicity endpoints in the data set.[15] However, this approach disregards the known toxicity information of the discarded compounds. Additionally, some studies applied imputation as a preprocessing step.[8,22–24] Imputation is a process that aims at 'completing' the missing data in the data set by substituting (in this case) the missing toxicity data with predicted values. Subsequently, multi-label classification methods are applied to the completed data set. Other studies applied multi-label classification methods directly to toxicity data sets without imputing the missing labels.[25]

Multi-label classification has not undergone comprehensive benchmarking for predictive toxicology. In the studies mentioned above different data sets, preprocessing steps, and evaluation metrics were used. Such inconsistent conditions make objective comparison and reproducibility of models' performances infeasible and complicate the process of selecting the best method for a particular purpose. Therefore, to support the progress in this area, we performed comprehensive benchmarking and analysis of 19,186 multi-label classification models generated using various combinations of computational methods. We evaluated the models' performances using several statistical metrics and analyzed their predictive performance in internal validation (fivefold cross-validation on the training set) and external validation (using a blind testing set), and across endpoints and compounds.

This review is organized into five sections. First, we explain single-label and multi-label classification. Second, we present an overview of different methods for multi-label classification used in the benchmark study. The methods were chosen either because they have been applied to multi-labeled toxicity data sets (i.e., data sets that include several toxicity endpoints), or they represent the state-of-the-art methods for multi-label classification. The third section describes the benchmarking process, the data set, and evaluation metrics. Next, we report the benchmarking analysis results. Finally, based on the results of the benchmark analysis, we provided recommendations for potential future research directions in this area.

SINGLE-LABEL AND MULTI-LABEL CLASSIFICATION

There are two main types of classification: single-label and multi-label classification. Single-label classification is used when each instance in a data set belongs to only one class[9] and is thus associated with only one label. Single-label classification can be binary (Figure 1(a)) if the data set can be split into only two classes (e.g., carcinogenic or noncarcinogenic compounds). If the data set can be divided into more than two classes, it is a multi-class classification[26] (Figure 1(b)). For example, compounds can be classified based on the degree of skin sensitization (e.g., high, moderate, low). In multi-class classification, classes are mutually exclusive. A given instance can belong to only one class.

In multi-label classification, however, an instance may belong to several classes simultaneously[9,27] (Figure 1(c)). In this case, each toxicity endpoint represents a label, and a given compound can have different activity for each endpoint. For example, if the data set contains two labels, carcinogenicity and genotoxicity, then a given compound can be only genotoxic, only carcinogenic, both genotoxic and carcinogenic, or both nongenotoxic and noncarcinogenic. In some cases, the labels are not known for some instances in a data set. For example, a compound can be genotoxic, but it is unknown if it is carcinogenic. This situation is referred to as 'missing labels,' and it is common in many multi-label classification datasets.

MODELING MULTI-LABEL TOXICITY DATA WITH MISSING LABELS

In this review, we considered three categories of computational methods to generate the predictive models as shown in Figure 2(a):

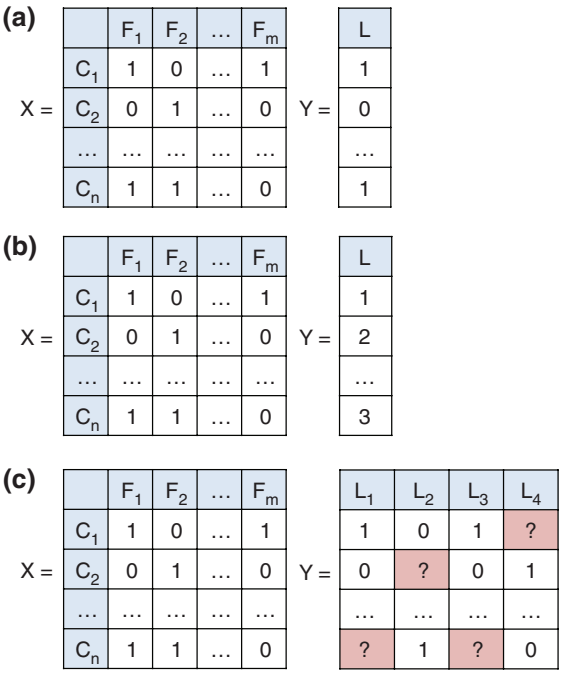

FIGURE 1 | Illustrations of single-label classification and multi-label classification. *X* is the data set in which feature vectors describe compounds C_1–C_n; *n* is the number of compounds; F_1–F_m are features; *m* is the number of features. *Y* is the label vector (in single-label classification) or the label matrix (in multi-label classification). (a) Binary classification. (b) Multi-class classification. (c) Multi-label classification. Missing labels are denoted with '?.' '1' and '0' are known labels.

- Nine methods for multi-label classification (Figure 2(a)).
- Each multi-label classification method is applied using one of nine base classifiers. Each base classifier is used with various computational settings including distance metrics, kernels, solvers, and splitting criteria when applicable (Figure 2(b)).
- Three methods for feature selection (Figure 2(c)).

Multi-Label Classification Methods

Here we provide a brief description of the multi-label classification methods while focusing on how to apply them to multi-label toxicity data sets with missing toxicity data. However, generic descriptions of these methods are provided in great detail in their corresponding references, which can be useful for readers in understanding how to apply the methods to related or similar types of datasets.

Binary relevance method requires generating one prediction model per label.[9] Each model is

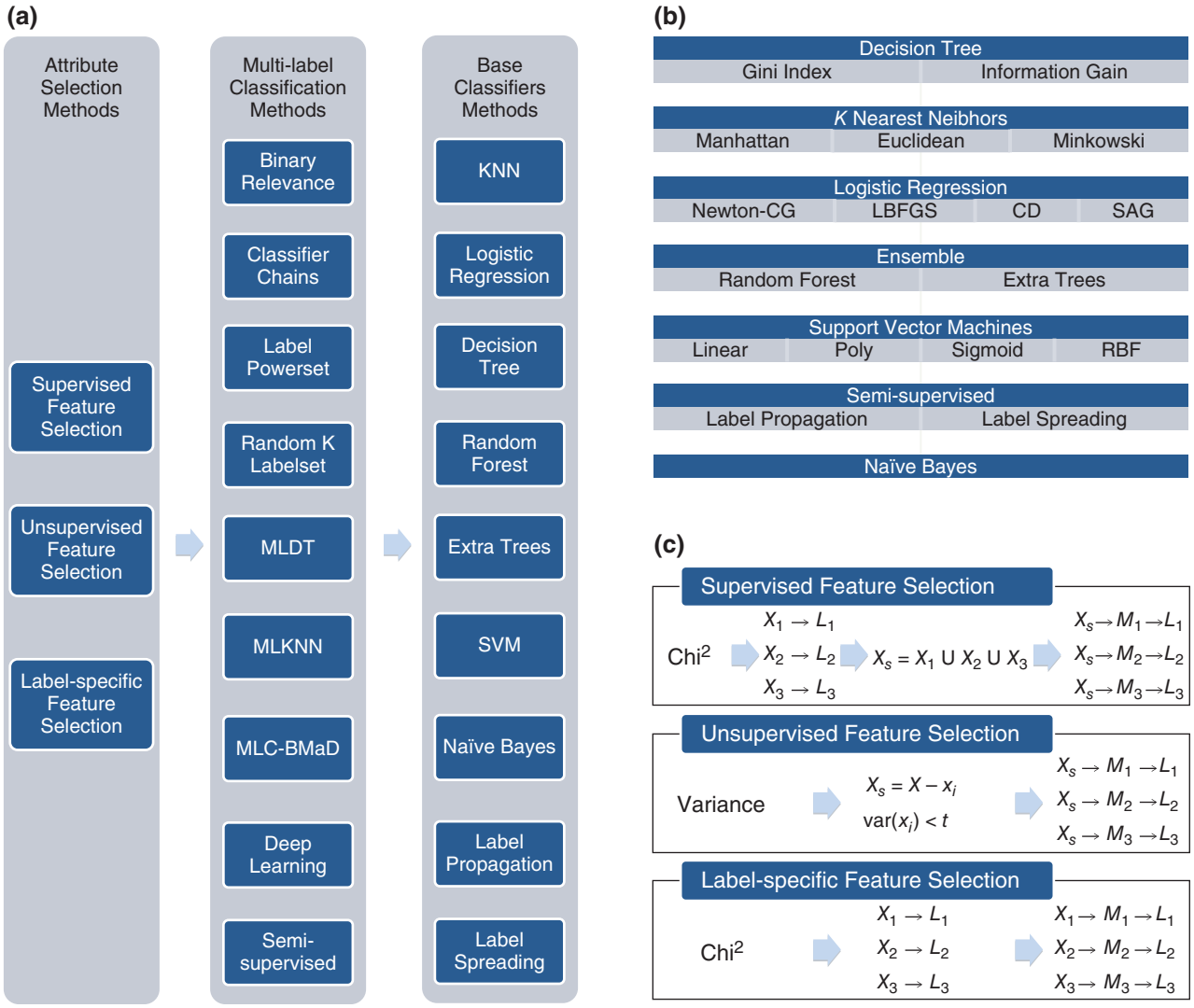

FIGURE 2 | Overview of modeling approaches. (a) Three categories of the computational methods including feature selection, multi-label classification, and base classifiers. MLDT, multi-label decision tree; MLKNN, multi-label *K* nearest neighbors; MLC-BMaD, multi-label Boolean matrix decomposition. (b) A list of base classifiers along with their corresponding kernels, solvers, splitting criteria, and distance metrics (when applicable). CD, Coordinate Decent; CG, Conjugate Gradient; LBFGS, Memory-limited quasi-Newton; SAG, Stochastic Average Gradient; RBF, Radial Basis Function. (c) Three feature selection methods. L_1, L_2, and L_3: labels; X: the original feature set; X_1, X_2, X_3: selected feature sets for labels L_1, L_2, and L_3, respectively; x_i: a single feature; X_s: the combined feature set; M_1, M_2, and M_3: models for endpoints L_1, L_2, and L_3, respectively; t: variance threshold.

produced using a base classifier (Base classifiers are detailed in the next section). Therefore, the number of generated base models is equal to the number of endpoints in the data set. For example, if the data set contains N toxicity endpoints, and a decision tree is used as a base classifier then N decision trees will be generated. Each trained decision tree will be used to predict a single toxicity endpoint of new compounds. However, the binary relevance method does not take into consideration correlations between endpoints.[28] To handle missing toxicity data only known toxicity data per endpoint is used to train base classifiers.

Classifier chains method is similar to binary relevance method since it requires generating one model per endpoint. However, it aims to identify relationships between endpoints by using some endpoints as features to predict other endpoints.[29] The workflow of this approach begins by selecting one endpoint randomly. A base classifier is trained using the known toxicity data of the endpoint. Secondly, the endpoint is used as a feature, and the missing data of the endpoint is imputed using the trained base classifier (Figure 3(c)). Next, a second endpoint is selected randomly, and the training set (which now includes

compounds features and the first endpoint) is used to train the second base classifier for modeling the second endpoint. These steps are repeated for all endpoints. If the data set contains N endpoints, N base classifiers will be trained. An application of this method to a multi-label dataset with missing data is described in reference.[25]

Label powerset method transforms multi-label classification to multi-class classification by treating combinations of labels as distinct classes (Figure 3 (d)). Only one multi-class base classifier is generated regardless of the number of endpoints in the data set. If the base-classifier does not support multi-class classification, there are two ways to transform multi-class classification to binary classification:

- One-versus-one approach creates a binary problem for each pair of classes.
- One-versus-rest (also called one-versus-all) approach creates a binary problem between one class, and the rest of the classes that are grouped to make the other class.

The label powerset approach aids in identifying correlations between endpoints. However, it may result in a large number of classes associated with a small number of compounds.[9] Typically, classes are encoded using combinations of the presence of toxicity. However, to handle the missing toxicity data, the classes are encoded using combinations of presence and absence of toxicity (Figure 3(d)). An application of this method to a multi-label data set with missing data is explained in reference.[25]

Random K labelset method is an ensemble approach that combines binary relevance and label powerset methods.[30] First, endpoints are randomly grouped into labelsets of length K ($1 \le K \le N$, where N is the number of endpoints in the data set). Then the label powerset method is applied to each labelset (Figure 3(e)). If $K = 1$ (i.e., each labelset contains only one endpoint), the approach resembles the binary relevance method. However, if $K = N$, where N is the number of endpoints (i.e., there is only one labelset that contains all the endpoints in the dataset), the approach resembles the label powerset method. There are two approaches to generate the labelsets: distinct and overlapping labelsets. The distinct labelsets approach generates disjoint labelsets such that each endpoint belongs to only one labelset. On the other hand, in the overlapping labelsets approach, labels can belong to more than one labelset. The overlapping labelset approach implements a voting mechanism to aggregate predictions for labels

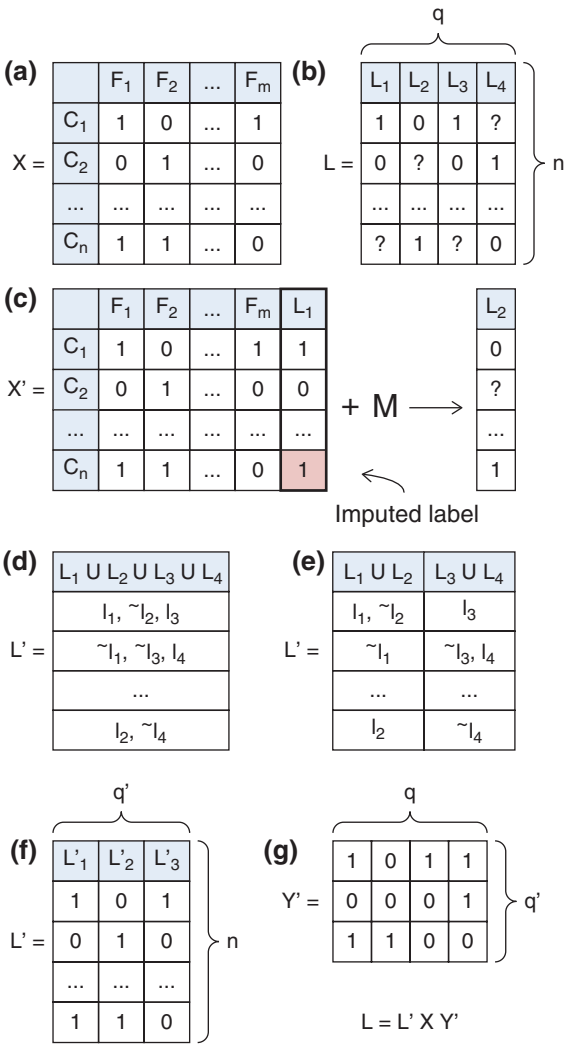

FIGURE 3 | Illustrations of some multi-label classification methods. (a) X is the matrix of features of compounds C_1–C_n, where n is the number of compounds, and their features F_1–F_m, where m is the number of features. (b) L is the label matrix that consists of four labels in this example. Positive and negative labels are denoted by '1' and '0', respectively, while '?' indicates missing labels. (c) Classifier chains method. Matrix X' consists of the feature matrix X from part (a) extended with the label L_1 from matrix L, where L is from part (b). The missing labels of L_1 are imputed. X' is used to train a model M to predict a second label, L_2. (d) Label powerset method. Matrix L' consists of the transformed multi-class labels. Each unique label combination is a distinct class. For example, l_1 indicates that L_1 is positive, while $\sim l_2$ indicates that L_2 is negative. Missing labels are not encoded. (e) Random K labelset method. Matrix L' consists of two labelsets of length $K = 2$, and each labelset is represented using the label powerset method. In this example, the first labelset consists of labels L_1 and L_2, and the second labelset consists of labels L_3 and L_4. (f) Multi-label Boolean matrix decomposition method. L' is the decomposed matrix that consists of three latent labels in this example: L'_1, L'_2, and L'_3. (g) Matrix Y' is the second matrix from the decomposition based on the multi-label Boolean matrix decomposition method.

that belong to several labelsets. In this review, we applied the distinct labelset approach. The main advantage of random *K* labelset method over the label powerset method is that grouping endpoints into labelsets can reduce the number of classes per labelset, and increase the number of compounds associated with each class. Decreasing the number of classes may reduce models complicity and consequently may improve their performance.[30]

Semi-supervised learning is applied when there is only a small proportion of data with known labels but a large proportion of data with missing labels, which is a common characteristic of multi-label data sets.[31] This approach uses both known and missing labels for model training. It was found that using a small data set with known labels in addition to data with missing labels may improve model's performance over merely using a small data set with known labels.[31] Algorithms that implement this method are explained in the next section.

Multi-label Boolean matrix decomposition method decomposes the labels matrix to a smaller matrix, which contains latent labels to encode endpoints correlations.[32] Then binary relevance or classifier chains approaches are applied to predict the latent labels (Figure 3(f)). Since the number of latent labels is smaller than the number of endpoints, fewer models are generated. This method aims to identify relationships between endpoints while reducing the number of generated models. To predict the toxicity of a new compound, models predict the latent labels. The predictions are transformed back to the original labels using the second matrix from the decomposition[32] as shown in Figure 3(g).

Deep learning method uses a neural network to model high-level abstractions of compound features using linear or nonlinear transformations. The transformed features may be more meaningful than the original features.[33] This approach uses binary relevance, classifier chains, label powerset, or random *K* labelset to generate the multi-label classification models. An application of deep learning to toxicity data with missing labels is available in Ref 17.

Multi-label decision tree is a modified algorithm of a binary or multi-class decision tree (explained in the next section), which can predict several toxicity endpoints at leaf nodes. It applies an extended definition of entropy by calculating average entropy across all endpoint.[34] This method implements implicit negativity to handle missing toxicity data (i.e., treat all missing data as negative/non-toxic).

Multi-label K nearest neighbor (KNN) is an extended algorithm of the lazy KNN algorithm (explained in the next section) to predict several toxicity endpoints.[35] However, it requires computing posterior and prior probabilities for each endpoint in the training set. Prior probability is the compound's likelihood to cause a toxicity effect, whereas posterior probability is the likelihood that some KNNs will cause the toxicity effect. This method applies a Bayesian rule for labels ranking.[35] To handle missing toxicity data, we implemented a modified version of this algorithm, so that only known data is used to calculate the prior and posterior probabilities.

Base Classifiers Methods

K Nearest Neighbors (KNN) algorithm is a lazy learning approach that uses a voting mechanism from the top *K* similar compounds to predict toxicity effects of new compounds.[36] This algorithm supports both binary and multi-class classifications. In applying this method, we considered two voting mechanisms: majority votes and weighted (by the distance) votes. When predicting the toxicity of a new compound, the majority votes mechanism assigns the class of majority of the nearest neighbor compounds to the new compound. Therefore, all votes from the nearest neighbor compounds have equal weights. However, the weighted votes mechanism considers the distance of the nearest neighbor compounds from the new compound. Therefore, votes of the nearest neighbor compounds that are further away from the new compound are weighted less. To calculate the distance between two compounds $\boldsymbol{p}$ and $\boldsymbol{q}$, where n is the number of features, we consider three distance metrics[37]:

$$Manhattan\ \ Distance: \|\boldsymbol{p}-\boldsymbol{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i| \quad (1)$$

$$Euclidean\ \ Distance: \|\boldsymbol{p}-\boldsymbol{q}\|_2 = \sqrt[2]{\sum_{i=1}^{n} |p_i - q_i|^2} \quad (2)$$

$$Minkowski\ \ Distance: \|p-q\|_x = \sqrt[x]{\sum_{i=1}^{n} |p_i - q_i|^x}, \quad where x \geq 1 \quad (3)$$

Logistic regression is a regression algorithm for binary classification that uses direct or iterative solvers for linear or nonlinear optimization problems.[38] We considered four solvers: Newton Conjugate Gradient,[39] Memory limited quasi-Newton (L-BFGS),[40] Coordinate Decent,[41] and Stochastic Average Descent.[42] Regularization is often used to prevent the solver's coefficients from overfitting by adding a regularization term. L1 regularization is calculated using the sum of the coefficients, but L2 regularization is calculated

using the sum of the square of the coefficients.[43] A multinomial fit function across the entire probability distribution is used in the case of multi-class classification.

Decision trees are a decision-making framework represented by a hierarchy of decision nodes. The top node in the hierarchy is called the root node. The data processed by the root node is split into two or more data subsets that are directed for processing to the nodes at the second level of the hierarchy. This process repeats for the remaining levels of the decision tree hierarchy. Nodes at the bottom of the hierarchy, which do not further split the data, are called leaf nodes. Compounds are classified by tracing a path from the root to leaf nodes. Homogeneous leaf nodes contain compounds that belong to one class. These are considered 'pure' nodes. Therefore, decision tree algorithms aim at creating decision trees with homogenous leaf nodes. In reality, this is rarely the case, and leaf nodes frequently contain mixed label data. We considered two metrics for measuring nodes impurity: information gain index, and Gini index (Raileanu & Stoffel, 2004). For the benchmarking analysis, the classification and regression trees algorithm[44] is used to create the decision trees.

An ensemble of trees algorithm generates several decision trees that process data in parallel. Then a voting mechanism is applied to aggregate predictions from all the trees. Two algorithms are considered in this review to create the ensembles: random forest and extra trees. Random forest algorithm selects random subsets from the training set then one decision tree is trained on each subset.[45] On the other hand, extra trees algorithm uses the whole data set, but the trees are generated randomly.[46]

Support Vector Machines (SVMs) *algorithm* works by finding a hyperplane to separate two classes of data. It uses kernel functions to transform the data to higher dimensions.[47] Let $\langle p, q\rangle$ denotes the dot product of two compounds p and q, let d be a polynomial degree, γ a coefficient, and r a constant. In this review, we considered four kernel functions[48]:

$$\text{Linear}: \langle p, q\rangle \tag{4}$$

$$\text{Polynomial}: (\gamma\langle p, q\rangle + r)^d \tag{5}$$

$$\text{Radial Basis Function (RBF)}: \exp\left(-\gamma|p-q|^2\right), \text{ where } \gamma > 0 \tag{6}$$

$$\text{Sigmoid}: \tanh(\gamma\langle p, q\rangle + r) \tag{7}$$

Naturally, SVM algorithm does not support multi-class classification. In this case, we considered the one-*vs*.-one approach for the benchmarking analysis. However, there is an extended definition of SVM to support multi-class classification.[49]

Label propagation and label spreading are two semi-supervised learning algorithms. Both algorithms work by constructing a similarity graph over all items in the data set.[50] Label spreading algorithm normalizes the weights of edges in the similarity graph, while label propagation does not modify the weights. We considered two kernels for propagating the labels: Radial Basis Function (RBF) and KNN.

Naïve Bayes is a probabilistic algorithm that uses prior probabilities of labels to calculate posterior probabilities of predicted labels with the naïve assumption of independence between pairs of features.[51] The prior probability of label L, P(L) is the number of co-associated with a label divided by the total number of compounds in the data set. The posterior probability P(L|C) of compound C is estimated as follows

$$P(L|C) \propto P(L)\prod_{i=1}^{n} P(x_i|L) \tag{8}$$

where n is the number of features and x_i is a feature of compound C.

Feature Selection Methods

Although data sets may contain features that are relevant to the toxicity endpoints, prediction models may not need all of them to produce good predictions. Feature selection is a preprocessing step aiming at eliminating features that do not contribute to models' predictive power. A reduced set of features may decrease models' complexity. Therefore, feature selection can improve models' interpretability, and in some cases, enhance their performance.[52] Feature selection is supervised if the labels are taken into consideration when selecting the features; otherwise, the feature selection is unsupervised. In the case of multi-label classification, it is more challenging to extract features that are relevant to all labels in the data set, especially in the cases with missing labels. We considered three methods for feature selection as summarized in Figure 2(c).

Supervised feature selection method uses statistical indices or scores to measure correlation or dependence between the features and the labels. This method removes features that are independent of the endpoints. In this approach, the top K dependent features per endpoint are selected then grouped to make the final feature set. Therefore, in this approach, only one feature set is used to train models to predict all endpoints in the data set. To handle missing toxicity

data the scores are calculated using only the known toxicity data per endpoint. In this review, we considered the Chi-square test, which ranks features according to their dependence to toxicity endpoints. The scores range from 0.0 (low dependence) to 1.0 (high dependence).[53]

Unsupervised feature selection method applies the variance score,[54] which removes features whose variance is less than a certain threshold. Features that have low variance scores (i.e., present or absent in all or most of compounds) may not be used to distinguish between toxic and nontoxic compounds. Unsupervised feature selection is suitable for data sets with missing toxicity data since labels are not involved in calculating the variance. However, the selected features may not contribute to models' predictive power since the features may not correlate with the labels. Similar to supervised feature selection, unsupervised feature selection generates one feature set for training models to predict all the endpoints.

Label-specific feature selection method is a supervised feature selection approach that aims at selecting features that are most suited for the multi-label classification method, which may help in increasing the model performance. Chi-square scores are used to measure correlations between the features and the labels as explained below:

- For the label powerset approach, the feature set is selected after generating the multi-class labels.
- For binary relevance and classifier chains methods, one set of features is selected per toxicity endpoint. Unlike supervised feature selection, label-specific feature selection does not group the selected features. Instead, each model is trained using the feature set specific to the considered endpoint. Therefore, the number of generated feature sets is equal to the number of toxicity endpoints in the data set. An example is provided in Figure 2(c).
- For the random K labelset approach, a set of features is selected for each labelset after generating the multi-class labels; then each model is trained using the feature set that corresponds to a particular labelset. Therefore, the number of generated feature sets is equal to the number of labelsets.

BENCHMARKING MODELS, DATA SET, AND EVALUATION METRICS

Generation of Models

Each model is generated using a combination of the methods mentioned above. As an example, one model is generated using the random K labelset method for multi-label classification with K equal to 2, with the base-classifier being a decision tree that uses information gain as a splitting criterion, and where supervised feature selection is used to select the best 16 features per label. The algorithm's parameters were tuned using fivefold cross-validation on the training set. Parameters for each method are detailed in Table S1, Supporting Information.

We generated more than 100,000 multi-label classification models. However, a majority of the models (~80,000) were useless since they predicted all compounds in internal validation as only positive or only negative. Thus, we excluded these models from the analysis. Here, we discuss the performance of 19,186 models (of which 1141 models are binary relevance models) that were able to provide meaningful predictions.

Figures S1 and S2 describe the characteristics of the 19,186 models. All relevant codes for training and testing the models, analyzing the results and creating the figures, and instructions how to run the code are available online at www.cbrc.kaust.edu.sa/mlc/index.php. Specifications of the used software are available in Appendix S1.

Data Set Description

Figure 4(a) shows toxicity profiles of 6644 pharmaceutical, environmental, and industrial compounds for 17 *in vivo* toxicity endpoints in five species. We compiled the data set from several public toxicity databases. A list of data sources is provided in Appendix S1. We gathered toxicity data with binary annotations [e.g., positive (toxic) or negative (nontoxic)]. We included endpoints that are associated with at least 50 toxic and 50 nontoxic compounds.[19] Only 17 endpoints from the used data sources satisfy these conditions as shown in Figure 4.

For each compound, the toxicity endpoints are marked as positive, negative, or missing (unknown toxicity). Figure 4(b) shows the number of toxic and nontoxic compounds associated with each toxicity endpoint. In this data set, a large number of compounds (3096 compounds) have known toxicity effects for genotoxicity in *Salmonella* toxicity endpoint. However, a small number of compounds (221 compounds) have known toxicity effects for developmental toxicity in mice. On average, each toxicity endpoint is associated with 1020 compounds. Some endpoints are balanced (i.e., associated with an equal number of toxic and nontoxic compounds), while other endpoints are imbalanced (i.e., associated with a large number of

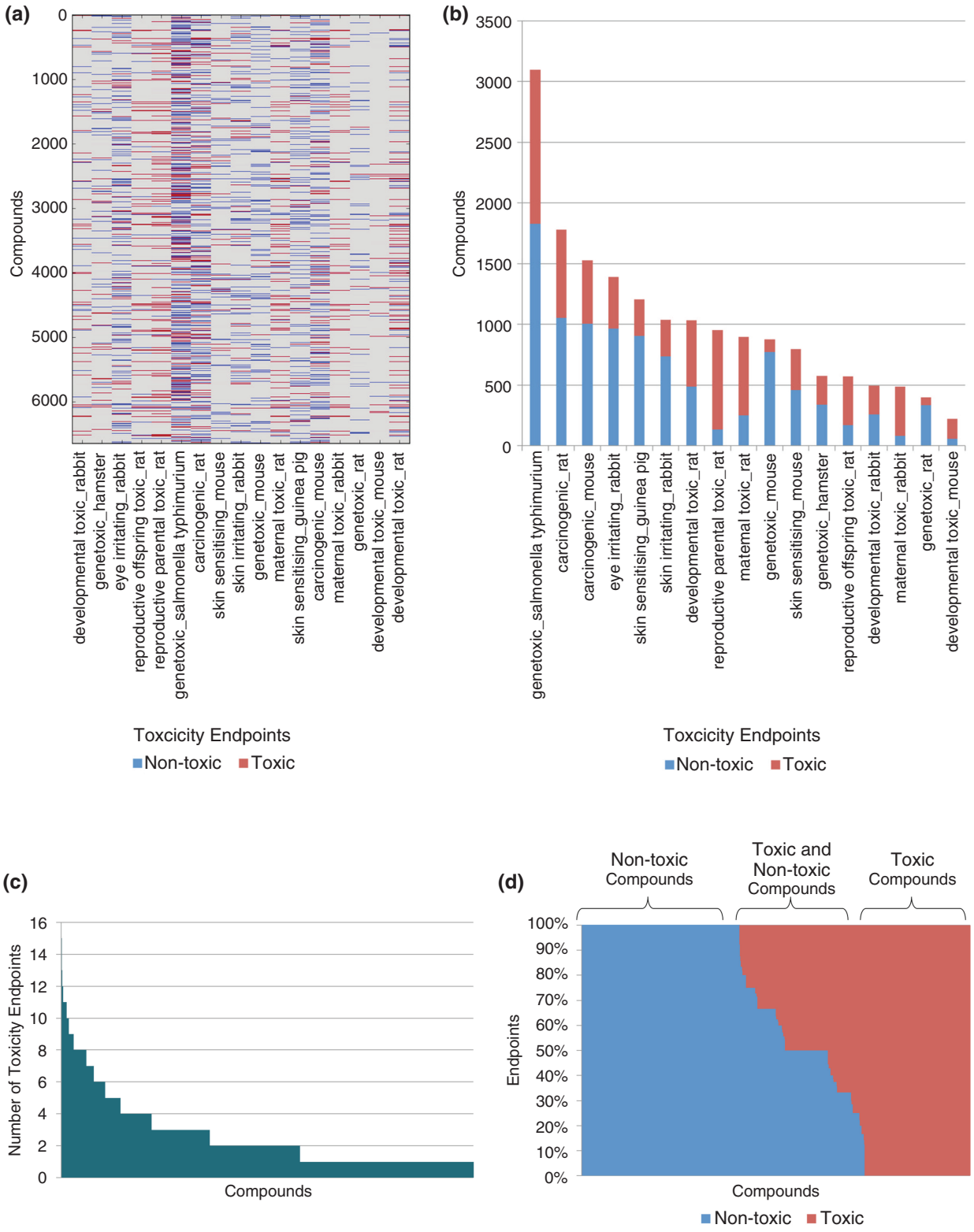

FIGURE 4 | Data set description. (a) Toxicity profiles of 6644 compounds for 17 toxicity endpoints. Each row corresponds to a compound, each column corresponds to a toxicity endpoint, and each cell represents a compound's activity per endpoint. Compounds are numbered from 0 to 6643. Red cells indicate active/toxic compounds, while blue cells indicate inactive/nontoxic compounds. However, gray cells denote the unknown toxicity. (b) A bar graph of the number of toxic and nontoxic compounds associated with each toxicity endpoint. (c) A bar graph of the number of known toxicity effects per compound. (d) A bar graph of the percentage of positive and negative toxicity effects per compound.

toxic compounds, but a small number of nontoxic compounds, or *vice versa*).

Some compounds cause several toxicity effects, and a given compound can be toxic for some endpoints, but non-toxic for others. In this data set, compounds have known toxicity effects (either positive or negative) for at least 1 and at most 15 endpoints (Figure 4(c)). In this data set, the average number of known toxicity effects per compound is 2.61. No compound has known toxicity effects for all the 17 endpoints. Figure 4(d) shows that 32% of compounds have both positive and negative effects. However, 41% of compounds have only known negative effects, while the remaining 27% have only known positive effects.

Eighty percentage of the compounds in the data set (5316 compounds) was used for training the models and internal validation using fivefold cross-validation, and the rest (1328 compounds) was reserved for external validation as a blind testing set. The data set is used without the imputation of the missing data. The final data set used for benchmarking and its description are available online at www.cbrc.kaust.edu.sa/mlc/index.php.

Compound Identification and Features

For each compound, we recorded CAS Registry Number, chemical name and synonyms (if available), simplified molecular-input line-entry system (SMILES) notation, and molecular formula. We used the SMILES notation for generating the compound's features. We used categorical and structural features that may provide insight into mechanisms of actions, including but not limited to:

- electrophilic, nucleophilic, and covalent reactivity mechanisms (e.g., Michael acceptors);
- radical mechanism by radical oxygen species formation;
- compound's potential to bind/interact with biological entities (e.g., DNA, proteins, peptides, or estrogen receptors);
- bioavailabity, biodegradation, bioaccumulation, and stability;
- functional groups (e.g., hydrocarbons; halogen; bases and acids groups; and oxygen, nitrogen, sulfur, phosphorus, or boron groups); and
- classes of compounds.

We used OECD QSAR toolbox,[13] which contains databases and computational tools to assign compounds to these categories. Moreover, structural features were generated using the OECD QSAR toolbox and PADEL toolbox[55] in Python 2.7. All the features are binary (e.g., present or absent). To reduce the number of features, we excluded features that are present in less than 5% of compounds. One hundred and eighty-six features satisfy this condition: 20 structural features from PADEL, 128 categorical features, and 38 structural features from the OECD QSAR toolbox. A complete list of all categorical and structural features is available in the dataset files.

Applicability Domain

The applicability domain of a prediction model is 'the theoretical space in which a model can make reliable predictions.'[56] We used Ambit Discovery tool[57,58] to define the applicability domain using the Euclidean distance to the Mean method. To determine the applicability domain, we entered the training compounds along with their generated 186 features described in the 'Compound identification and features' section above. After defining the domain, we used the tool to determine wither the test compounds fall within or outside the applicability domain. Only two test compounds fall outside the applicability domain.

Performance Evaluation Metrics

Predictions were classified into four categories: True Positive (TP: compounds are toxic and predicted to be toxic), True Negative (TN: compounds are nontoxic and predicted to be nontoxic), False Positive (FP: compounds are nontoxic but predicted to be toxic), and False Negative (FN: compounds are toxic but predicted to be nontoxic). Model macro-average performance was determined by calculating the performance per endpoint then averaged across all endpoints. The performance metrics are the following[53]:

- Recall (also called sensitivity) is the proportion of toxic compounds that are predicted to be toxic. It is defined as TP/(TP + FN).
- Specificity is the proportion of nontoxic compounds that are predicted to be nontoxic. It is defined as TN/(TN + FP).
- Precision (also called positive predictive value) is the proportion of correctly predicted toxic compounds out of all compounds that are predicted to be toxic. It is defined as TP/(TP + FP).
- Negative predictive value (NPV) is the proportion of correctly predicted nontoxic compounds out of all compounds that are predicted to be nontoxic. It is defined as TN/(TN + FN).

- F1-score is the harmonic mean of precision and recall. It is defined as 2*(Precision*Recall)/(Precision + Recall).
- Accuracy is the success rate in predicting toxic and nontoxic compounds. It is defined as (TP + TN)/(TP + FP + TN + FN).
- Area Under Receiver Operating Characteristics curve (AUROC) is the area under a curve representing the relationship between TP rate (i.e., recall) and FP rate (i.e., 1 – specificity) for a model.

The above metrics produce scores between 0.0 and 1.0. A perfect model would score 1.0 for all metrics. High specificity but low recall scores indicate that models predicted most of the compounds as nontoxic. On the other hand, high recall but low specificity suggests that models predicted most of the compounds as toxic. Additionally, high F1-score is achieved when both precision and recall are high.

Accuracy scores show the average performance in predicting toxic and nontoxic compounds for balanced data sets. However, in the case of imbalanced data sets (e.g., the number of toxic compounds is much larger than nontoxic compounds or *vice versa*), accuracy scores can be overestimated if the models adjust to the majority class (e.g., predict all compounds as toxic where toxic compounds represent the majority class). However, the AUROC metric is not sensitive to imbalanced data sets.

BENCHMARKING ANALYSIS

Multi-Label Versus Binary Relevance Models Macro-Average Performance

Binary relevance method, which works by generating an individual model for each label, is often used as a baseline to assess the performance of other multi-label classification methods and determine whether using endpoints correlations enhances model's performance.[9] Figure 5 shows the models macro-average performances based on five metrics: accuracy, F1-score, precision, recall, and specificity in internal (Figure 5(a)) and external (Figure 5(b)) validations. The gray areas in bar graphs show the performance range of binary relevance models.

The performance of many multi-label models falls within the performance range of binary relevance models (i.e., their performances fall within the gray areas in Figure 5). However, some multi-label models, such as those generated by random *K* labelset and label powerset, significantly exceed the performance of binary relevance models (i.e., their performances fall above the gray areas in Figure 5). On the other hand, other multi-label models, such as those generated by deep learning and multi-label Boolean matrix decomposition methods, appear to be considerably weaker (i.e., their performances fall below the gray areas in Figure 5). Moreover, some multi-label models outperformed binary relevance models across all the five performance metrics (e.g., leftmost models in bar graphs in Figure 5(a)). However, some multi-label models outperformed binary relevance models in only one performance metric (e.g., middle to rightmost models in Figure 5(a) outperformed binary relevance models in specificity only).

Internal Versus External Validation

We analyzed variability in predictive performance between internal and external validations. Figure 6 shows scatter plots of macro-average performances of models in internal and external validation based on the five performance metrics. Properly fitted models have similar performance in internal and external validation and appear close to the diagonal (from (0,0) point to (1) point) of the scatter plots (marked in green in Figure 6). We define the 'diagonal region' as the region of properly fitted models where the difference between the performances in external and internal validations is <20%. Overfitted models achieved high performance in internal validation, but weaker performance in external validation.[59] These models appear below the diagonal region (marked in orange in Figure 6). However, poor performance in internal validation, but high performance in external validation, may occur when the characteristics of the data in the testing set are similar to the portion of the data in the training set were the models performed well. These models appear above the diagonal region (marked in blue in Figure 6).

The majority of the generated models have similar performances in internal and external validation. Overall, 4.13, 6.54, 8.73, 6.84, and 2.92% of models show big differences (≥ 20%) between internal and external validation in accuracy, F1-score, precision, recall, and specificity, respectively. Table 1 details the proportion of models that appear above, within or below the diagonal region generated by each multi-label classification method. Notably, the performance of all models generated by multi-label *K* nearest neighbors, binary relevance, and classifier chains methods reside within the diagonal region. Similarly, the majority of models generated by deep learning, semi-supervised learning, multi-label Boolean matrix

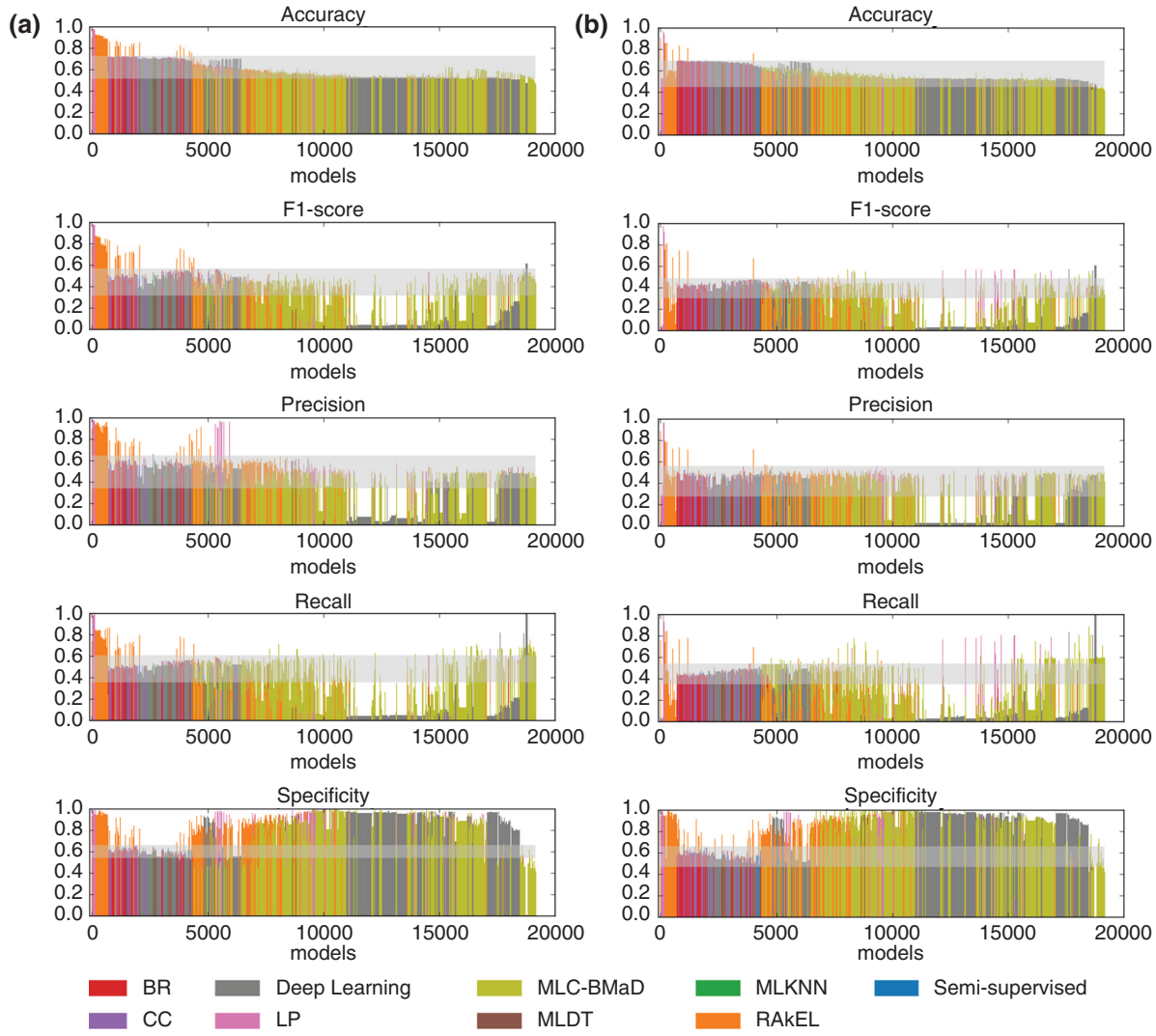

FIGURE 5 | Comparison of macro-average performances of multi-label and binary relevance models in (a) internal and (b) external validation. Bar graphs show models performance via five metrics: accuracy, F1-score, precision, recall, and specificity. Models are numbered from 0 to 19,185. The gray areas in bar graphs show the performance range of binary relevance models. BR, binary relevance; CC, classifier chains; LP, label powerset; MLC-BMad, multi-label Boolean matrix decomposition; MLDT, multi-label decision tree; MLKNN, multi-label *K* nearest neighbors; RAkEL: random *K* labelset.

decomposition methods reside in the diagonal region. However, the performance of a significant percentage models generated by random *K* labelset and label powerset methods reside below the diagonal region (i.e., overfitted). Additionally, many models generated by the label powerset method reside above the diagonal region.

Best Performing Models

To determine the best-performing models, each model was assigned seven ranks according to its average performance in internal and external validation using seven performance metrics: accuracy, F1-score, precision, recall, AUROC, specificity, and NPV. The seven ranks were averaged to calculate the final rank. Figure 7 shows the accuracy scores per endpoint of the top-ranked models generated by each multi-label classification method and the top-ranked model generated by the binary relevance method.

Description of the top 10 ranked models generated by each multi-label classification method and binary relevance method is provided in Table S2 along with their macro-average performance calculated using the seven performance metrics. Models' ranks are provided in Table S3 along with their average internal and external validation performances calculated using the seven performance metrics.

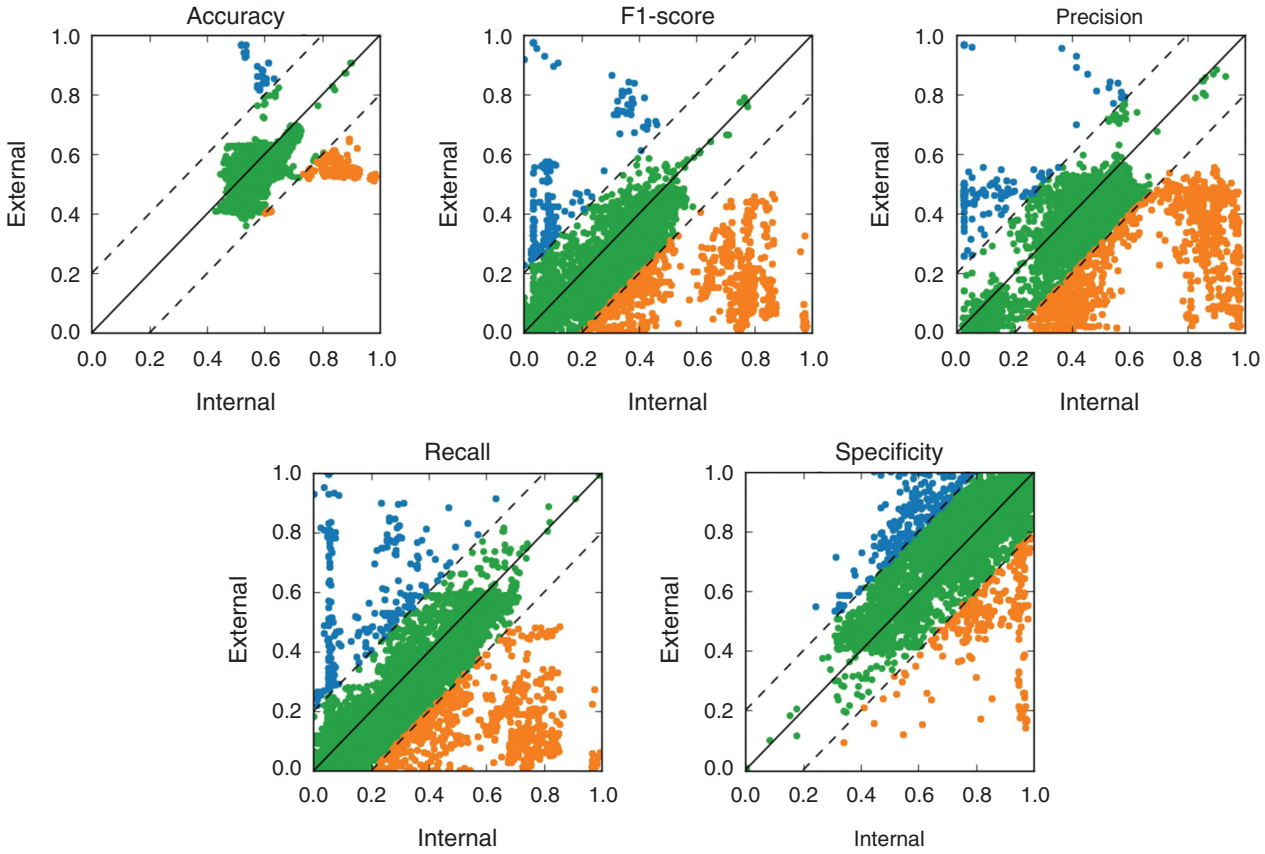

FIGURE 6 | Comparison of macro-average performances of models in internal and external validations. The scatter plots demonstrate models performances via five metrics: accuracy, F1-score, precision, recall, and specificity. The *x*-axis and *y*-axis show model performances in internal and external validation, respectively. The closer the models are to the diagonal (from (0,0) point to (1) point)) of the scatter plots, the more similar is their performance in internal and external validations. However, models that have high variability between internal and external performance appear below or above the diagonal region and are marked in orange and blue, respectively.

Performances and ranks of all 19,186 models are available at www.cbrc.kaust.edu.sa/mlc/index.php.

The results in Table S2 and Figure 7 indicate that random *K* labelset method considerably outperformed binary relevance method. The top-ranked random *K* labelset model achieved macro-average accuracy, F1-score, precision, recall, AUROC, specificity, and NPV scores of 90, 76, 86, 74, 83, 92, and 91%, respectively, in internal validation, and 91, 79, 86, 76, 85, 94, and 92%, respectively, in external validation. On the other hand, the top-ranked binary relevance model achieved for the above-mentioned measures 70, 56, 56, 57, 57, 58, and 67%, respectively, in internal validation, and 65, 48, 48, 50, 51, 53, and 53%, respectively, in external validation. This indicates that utilizing labels correlations by the random *K* labelset method improved the performance.

The top-performing models generated by label powerset method exhibited overfitting. For example, the top-ranked label powerset model achieved for the above-mentioned measures 98, 97, 97, 98, 98, 98, and 98%, respectively, in internal validation, and 53, 5, 46, 4, 50, 97, and 53%, respectively, in external validation. Overfitting is possible side effect of the label powerset method, since this method tends to generate a large number of classes that are associated with a small number of compounds. The random *K* labelset method aims to avoid this problem by grouping the labels into labelsets, which can reduce the number of generated classes per labelset.

Moreover, classifier chains and multi-label *K* nearest neighbors slightly outperformed the binary relevance models. The multi-label *K* nearest neighbor method depends on calculating prior and posterior probabilities of the labels, which could be difficult to estimate accurately for multi-label data sets with missing labels. In addition, one possible factor that could affect the performance of the classifier chains method is the order in selecting the endpoints. This is done randomly in many implementations of this method. Also, imputation of missing labels is necessary when the classifier chains method is applied to

TABLE 1 | The Proportion of Models that Have High or Low Variability in Predictive Performance between Internal and External Validations Based on Five Metrics

Multi-Label Classification Method	Scatter Plot Region	Proportion of Models Per Metric[1]				
		ACC	F1	RR	RE	SP
Random *K* labelset	Above diagonal region	1.25%	2.73%	1.76%	4.26%	1.06%
	Within diagonal region	70.54%	63.83%	66.47%	62.95%	90.80%
	Below diagonal region	28.21%	33.44%	31.78%	32.79%	8.14%
Label powerset	Above diagonal region	2.24%	22.72%	23.92%	20.03%	0.30%
	Within diagonal region	81.02%	53.66%	49.18%	56.65%	81.46%
	Below diagonal region	16.74%	23.62%	26.91%	23.32%	18.24%
Multi-label *K* nearest neighbors	Above diagonal region	0.00%	0.00%	0.00%	0.00%	0.00%
	Within diagonal region	100.00%	100.00%	100.00%	100.00%	100.00%
	Below diagonal region	0.00%	0.00%	0.00%	0.00%	0.00%
Deep learning	Above diagonal region	0.00%	0.00%	0.00%	0.00%	0.00%
	Within diagonal region	100.00%	100.00%	98.22%	100.00%	100.00%
	Below diagonal region	0.00%	0.00%	1.78%	0.00%	0.00%
Binary relevance	Above diagonal region	0.00%	0.00%	0.00%	0.00%	0.00%
	Within diagonal region	100.00%	100.00%	100.00%	100.00%	100.00%
	Below diagonal region	0.00%	0.00%	0.00%	0.00%	0.00%
Classifier Chains	Above diagonal region	0.00%	0.00%	0.00%	0.00%	0.00%
	Within diagonal region	100.00%	100.00%	100.00%	100.00%	100.00%
	Below diagonal region	0.00%	0.00%	0.00%	0.00%	0.00%
Semi-supervised Learning	Above diagonal region	0.00%	0.00%	0.00%	1.14%	0.00%
	Within diagonal region	100.00%	99.43%	99.43%	98.86%	97.16%
	Below diagonal region	0.00%	0.57%	0.57%	0.00%	2.84%
Multi-label decision tree	Above diagonal region	0.00%	0.00%	0.00%	0.00%	0.00%
	Within diagonal region	100.00%	100.00%	45.00%	100.00%	100.00%
	Below diagonal region	0.00%	0.00%	55.00%	0.00%	0.00%
Multi-label Boolean matrix decomposition	Above diagonal region	0.00%	0.00%	0.01%	0.52%	2.58%
	Within diagonal region	99.62%	97.56%	93.26%	96.78%	96.82%
	Below diagonal region	0.38%	2.44%	6.72%	2.70%	0.60%

ACC, accuracy; F1, F1-score; PR, precision; RE, recall; SP, specificity.
[1] The percentage of generated models whose performance falls above, within, or below the diagonal region according to each performance metric. For example, the first row shows that 1.25% of models generated by random *K* Labelset method reside above the diagonal region according to their accuracy scores. Similarly, only 2.73% of models created by random *K* labelset method reside above the diagonal region according to their F1-scores.

data sets with missing labels. The accuracy of the imputation may affect the performance of the models.

However, multi-label Boolean matrix decomposition, semi-supervised learning, and deep learning performed worse than the top-ranked binary relevance models. Deep learning works by converting the original features into new features that my not always be sufficiently useful for the intended application. Moreover, the AUROC curve scores of the top-performing multi-label Boolean matrix decomposition range between 49 and 52% (Table S2), which is close to random predictions. Additionally, the semi-supervised learning method does not take label-correlation into consideration, which may resulted in its poor performance.

Notably, the top-ranked multi-label decision tree models achieved high specificity but low recall scores. For example, the top-ranked multi-label decision tree model achieved specificity, recall, precision, and NPV scores of 94, 11, 61, and 54%, respectively, in internal validation, and 95, 6, 47, and 53%, respectively, in external validation. This suggests that these models predicted most of the compounds as negative with low accuracy. This performance may be due to the implementation of

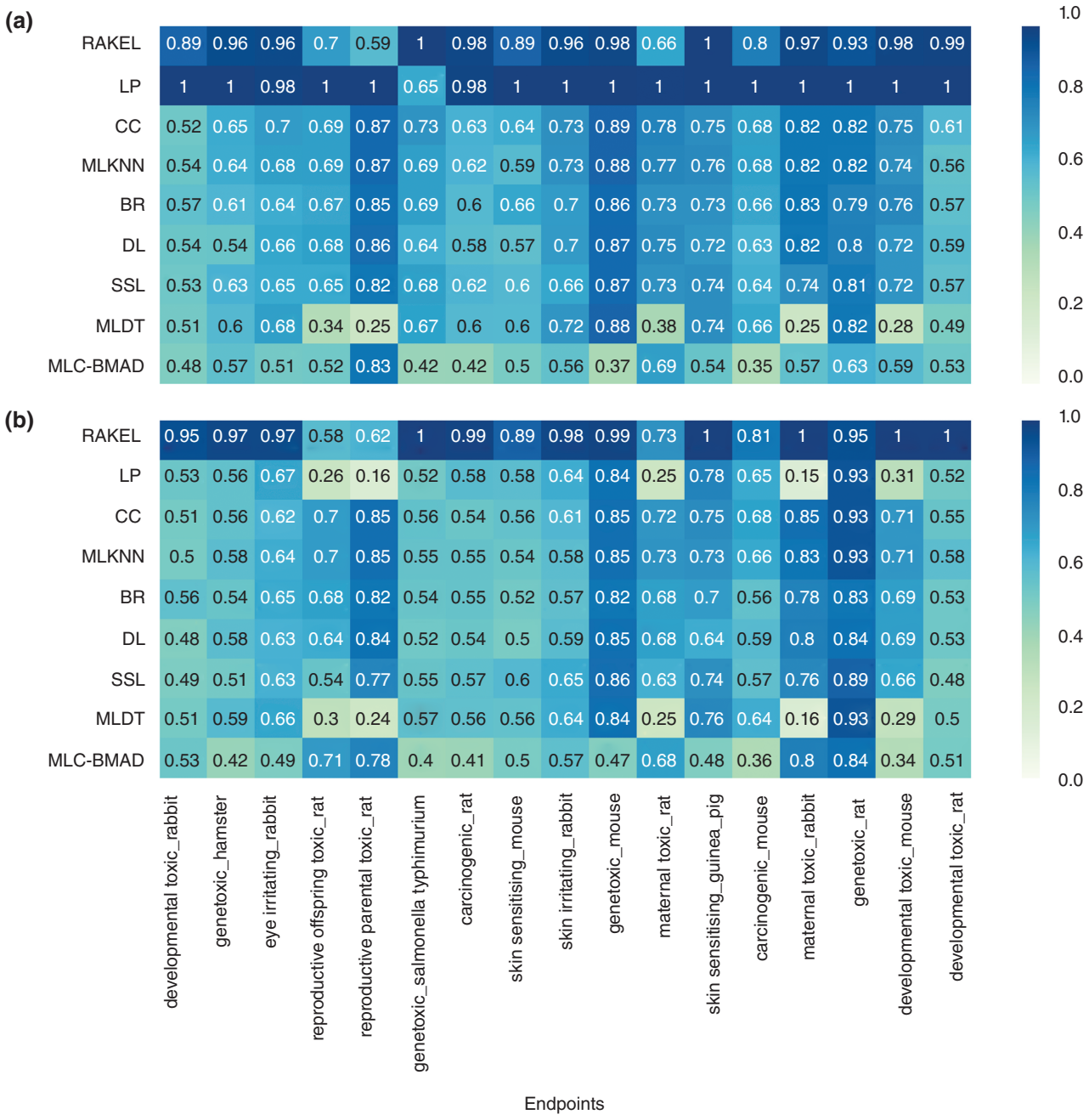

FIGURE 7 | Accuracy scores per endpoint of the top-ranked models generated by each multi-label classification method and top ranked binary relevance model in (a) internal and (b) external validation. Rows correspond to the multi-label classification methods and the binary relevance method. Column corresponds to endpoints. Each cell shows the accuracy scores of each method per endpoint. The scores range from 0.0 (worst performance) to 1.0 (best performance). BR, binary relevance; CC, classifier chains; DL, deep learning; LP, label powerset; MLC-BMaD, multi-label Boolean matrix decomposition; MLDT, multi-label decision tree; MLKNN, multi-label *K* nearest neighbor; RAkEL, random *K* labelset; SSL, semi-supervised learning.

multi-label decision tree method that assumes implicit negativity (i.e., treat all missing labels as negative).

Moreover, we identified seven multi-label models that exceeded the performance range of binary relevance models in all five metrics: accuracy, F1-score, precision, recall, and specificity in both internal and external validation. Then, we ranked the seven models as explained above. The seven models were generated using random *K* labelset method. Description of these seven models is provided in

Table 2 along with their calculated seven performance metrics. Models' ranks are provided in Table S4 along with their average internal and external validation performances calculated by the seven performance metrics.

In Table 2, the best-performing model is generated by random K labelset approach (with $K = 4$), decision trees as base classifiers, and label-specific feature selection method. The length of the labelset K is four indicating that each labelset has four endpoints. Since there are 17 endpoints, the method generated five labelsets. Each labelset includes four endpoints except for one labelset, which contains only one endpoint. The model created five decision trees (one tree for each labelset) using the Gini index for measuring nodes purity. The parameters of the decision trees were tuned using fivefold cross-validation on the training set. Each tree was trained using 16 features selected by the label-specific feature selection method. The model achieved accuracy, F1-score, precision, recall, AUROC, specificity, and NPV scores of 90, 76, 86, 74, 83, 92, and 91%, respectively in internal validation, and 91, 79, 86, 76, 85, 94, and 92%, respectively, in external validation.

Random Chance Analysis

Figure 8 shows AUROC scores of the top-ranked models generated by each multi-label classification method and the binary relevance method. AUROC scores range from 0.0 (worst performance) to 1.0 (best performance), and a score of 0.5 indicates random predictions. Figure 8 shows that the binary relevance method performed slightly better than random. However, the random K labelset model achieved high AUROC scores in predicting most of the endpoints. Interestingly, the label powerset method achieved high AUROC scores in internal validation, but its performance is almost random in external validation.

Estimating Toxicity of a Given Endpoint Using Average Toxicity Values of Other Endpoints

We investigated the possibility to estimate the toxicity of a given compound for a specific endpoint using the average toxicity measurements of other endpoints for the same compound. Since this approach requires calculating the average toxicity measurements of the endpoints, we applied this approach only to compounds that have known toxicity data for at least two endpoints. The macro-average accuracy, F1-score, precision, recall, AUROC, specificity, and NPV scores in internal validation are 63, 57, 57, 62, 62, 63, and 64%, respectively. Moreover, in external validation, the previously mentioned scores are 60, 51, 52, 54, 57, 58, and 60%, respectively. These results indicate that this approach achieved worse than the binary relevance method on the data set we used. Additionally, Figure 9 shows the performance of this approach per endpoint.

However, it should be noted that these results are not comparable to other results in this review since this approach is applied to a subset of the data set. Moreover, there are two main issues with this approach. First, it is inapplicable to new compounds for which there is no toxicity information at all. In this review, we treat the compounds in the testing set as if they are new compounds. We use the models, which were trained using the training set, to predict all toxicity endpoints in the testing set. Then, we use the known toxicity information of the compounds in the testing set to verify the predictions. Second, some compounds in this data set have known toxicity information for only one endpoint, so there is not enough data to calculate 'average' experimental values.

Predictability of Endpoints

We analyzed variability in models' predictive performances across endpoints. The heat maps in Figure 10 show models' performance per endpoint in internal (Figure 10(a)) and external (Figure 10(b)) validation. We used mean absolute error metric (MAE),[60] which ranges from 0.0 (perfect performance) to 1.0 (worst performance), to compare predictions per endpoint with the true labels. Let y and $\hat{y}$ denote the vector of true labels (i.e. the gold standard) and the vector of predicted labels, respectively, and let n be the number of samples. The MAE is defined as[60]

$$MAE(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{9}$$

We used hierarchal clustering[61] to group endpoints according to models' performances into two clusters of endpoints with high predictability and endpoints with low predictability. Hierarchal clustering is detailed in Ref 61. Briefly, a hierarchal clustering algorithm works by grouping data into a hierarchy of clusters visualized via a dendrogram. The height of each bar in the dendrogram represents the distance between the clusters or instances. A short bar in the dendrogram indicates small distance (i.e., high similarity), while a tall bar indicates large distance

TABLE 2 | Description and Macro-Average Performance of the Best Seven Performing Models

Number	Method[1]	Base Classifier	Classifier Setting[2]	Feature Selection Method	Number of Features[3]	K[4]		ACC	F1	RR	RE	AUROC	SP	NPV
1	Random *K* labelset	Decision Tree	Gini Index	LSFS	16	4	CV*	0.90	0.76	0.86	0.74	0.83	0.92	0.91
							Test	0.91	0.79	0.86	0.76	0.85	0.94	0.92
2	Random *K* labelset	Decision Tree	Information Gain	LSFS	16	4	CV*	0.90	0.75	0.85	0.73	0.83	0.94	0.91
							Test	0.91	0.77	0.85	0.75	0.84	0.94	0.92
3	Random *K* labelset	Random Forest	Gini Index	UFS	25	3	CV*	0.88	0.78	0.90	0.74	0.82	0.90	0.89
							Test	0.87	0.76	0.88	0.72	0.81	0.91	0.87
4	Random *K* labelset	Random Forest	Gini Index	SFS	24	3	CV*	0.88	0.77	0.89	0.74	0.81	0.89	0.88
							Test	0.87	0.77	0.87	0.75	0.83	0.90	0.88
5	Random *K* labelset	Random Forest	Information Gain	LSFS	2	2	CV*	0.84	0.65	0.85	0.63	0.74	0.84	0.85
							Test	0.83	0.64	0.85	0.63	0.74	0.85	0.83
6	Random *K* Labelset	Random Forest	Information Gain	UFS	25	2	CV*	0.84	0.71	0.83	0.69	0.74	0.80	0.87
							Test	0.82	0.66	0.79	0.64	0.72	0.80	0.82
7	Random *K* Labelset	Naïve Bayes	—	SFS	87	2	CV*	0.78	0.70	0.70	0.71	0.72	0.73	0.76
							Test	0.76	0.67	0.67	0.67	0.71	0.74	0.74

ACC, accuracy; F1, F1-score; PR, precision; RE, recall; AUROC, area under Receiver Operating Characteristics curve; SP, specificity; NPV, negative predictive value; CV, cross-validation (internal validation); Test, testing set (external validation); SFS, Supervised feature selection; UFS, Unsupervised feature selection; LSFS, Label-specific feature selection.

[1] Multi-label classification approach.

[2] Settings used for the base classifier. For decision trees, it is the metric for measuring nodes purity.

[3] In the case of LSFS, this is the number of features selected per endpoint. In SFS and UFS, this is the total number of selected features.

[4] The size of the labelset.

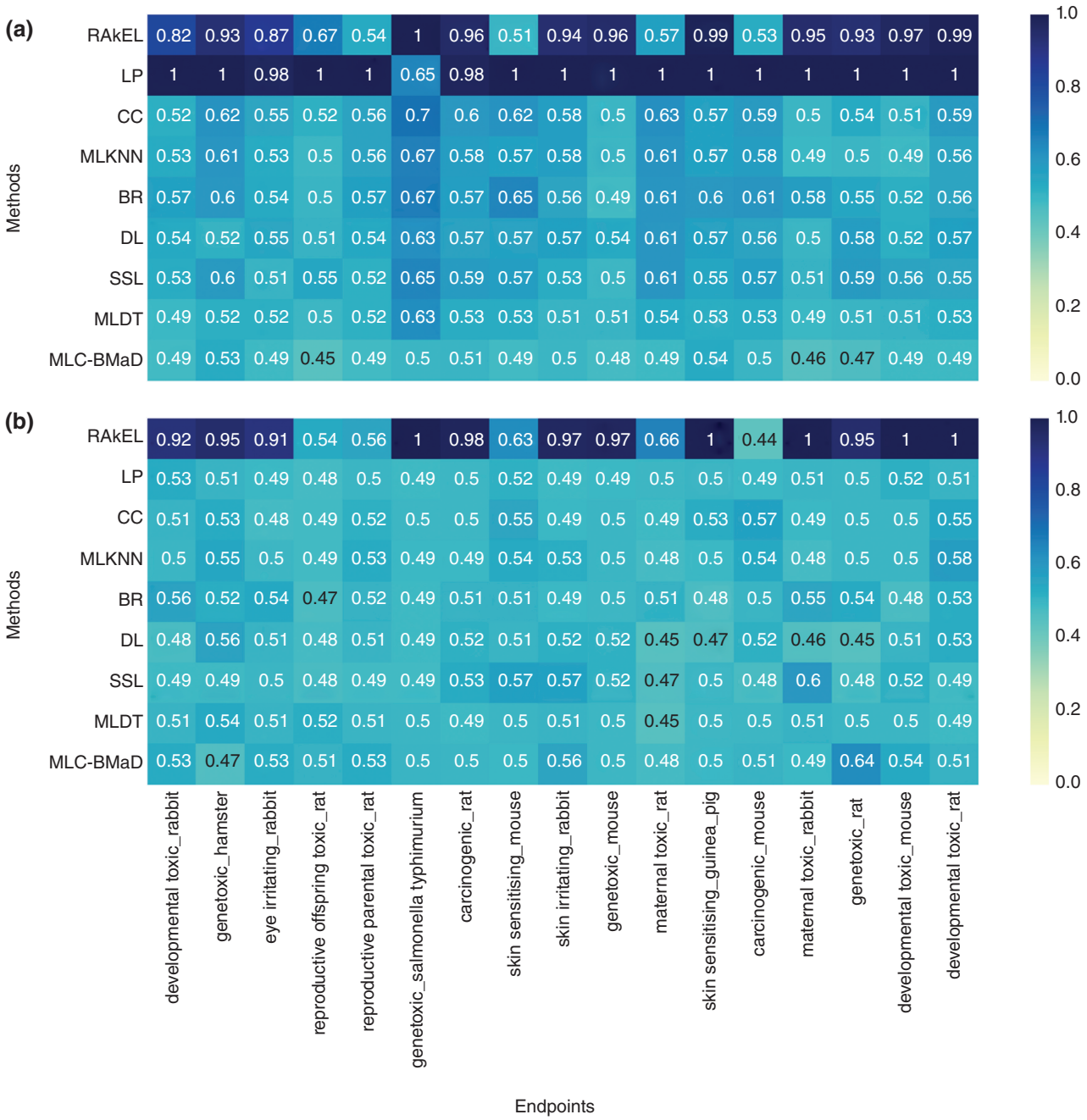

FIGURE 8 | Area Under Receiver Operating Characteristics curve (AUROC) scores of the top-ranked models generated by each multi-label classification method and the binary relevance method per endpoint in (a) internal and (b) external validation. Rows correspond to the multi-label classification methods and the binary relevance method. Column corresponds to endpoints. Each cell shows the AUROC scores of each method per endpoint. The scores range from 0.0 (worst performance) to 1.0 (best performance). AUROC scores of 0.5 indicate random predictions. BR, binary relevance; CC, classifier chains; DL, deep learning; LP, label powerset; MLC-BMaD, multi-label Boolean matrix decomposition; MLDT, multi-label decision tree; MLKNN, multi-label K nearest neighbor; RAkEL, random K labelset; SSL, semi-supervised learning.

(i.e., low similarity). We used the correlation distance[37,53] to calculate the distance between pairs of endpoints. The correlation distance between two endpoints u and v is defined as

$$1-\frac{(u-\overline{u})\cdot(v-\overline{v})}{\|u-\overline{u}\|_2\|v-\overline{v}\|_2} \tag{10}$$

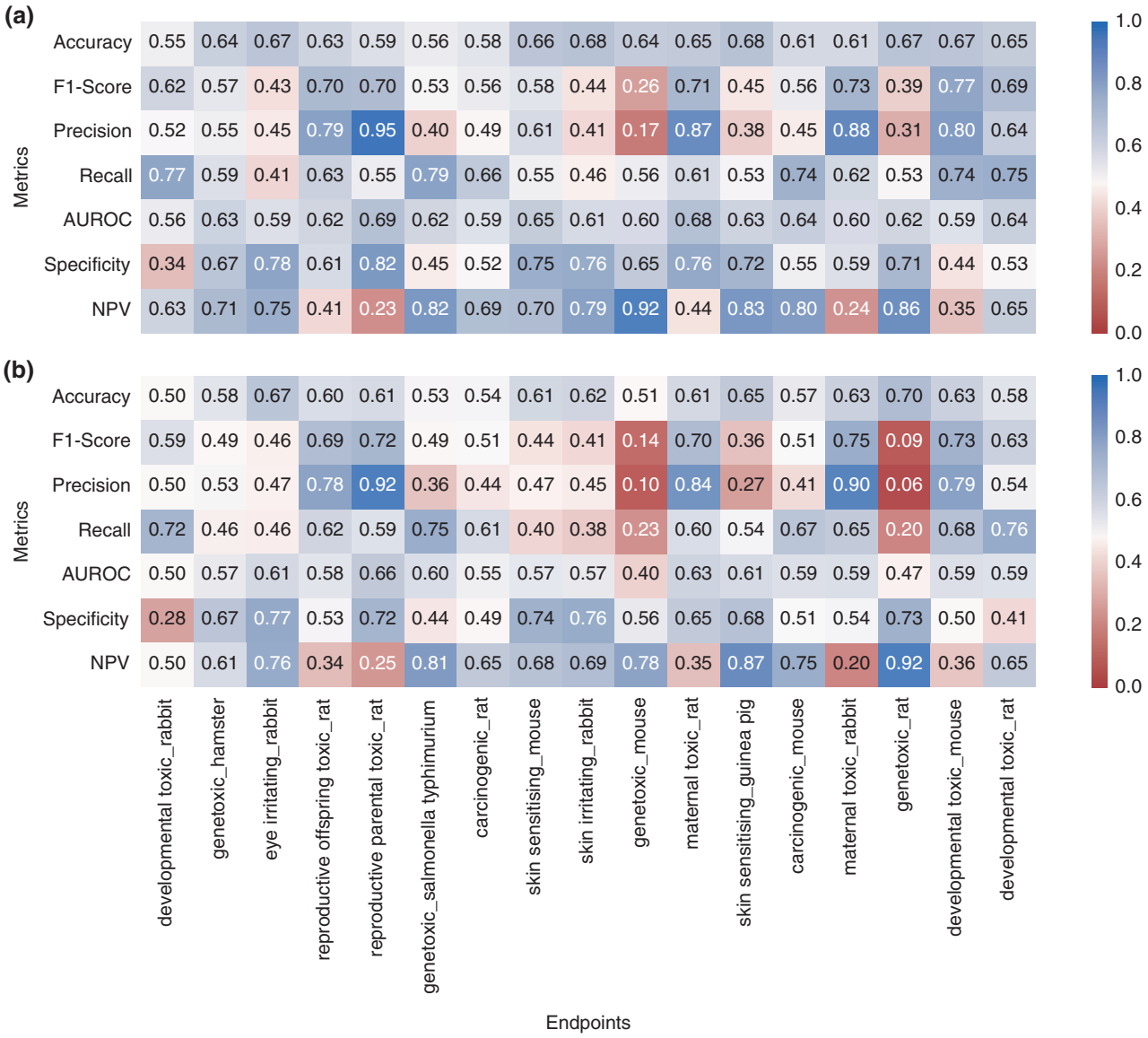

FIGURE 9 | Performance of estimating the toxicity of a given endpoint using average toxicity values of other endpoints in (a) internal and (b) external validation. Each row corresponds to a performance metric, and each column corresponds to an endpoint. Each cell shows the calculated scores per endpoint. The scores range from 0.0 (worst performance) to 1.0 (best performance).

where $\overline{u}$ and $\overline{v}$ are the means of u and v, respectively. The distance between pairs of clusters is calculated using the average linkage method, which is the average distance between pairs of endpoints from each cluster.

We consider endpoints to have high predictability when a majority of the models achieve high performance in predicting them; otherwise, the endpoints have low predictability. The clusters marked as green in dendrograms in Figure 10 contain 11 (Figure 10(a)) and 12 (Figure 10(b)) endpoints that have high predictability in internal (Figure 10(a)) and external (Figure 10(b)) validations, respectively. However, the clusters marked as orange in dendrograms contain six (Figure 10(a)) and five (Figure 10(b)) endpoints that have low predictability in internal (Figure 10(a)) and external (Figure 10(b)) validations, respectively. These observations suggest that it is harder to predict some endpoints than the others. Nonetheless, some models achieved high performance in predicting endpoints with low predictability.

Predictability of Compounds

Similarly, we analyzed the predictive performance of the models across compounds. The heat maps in

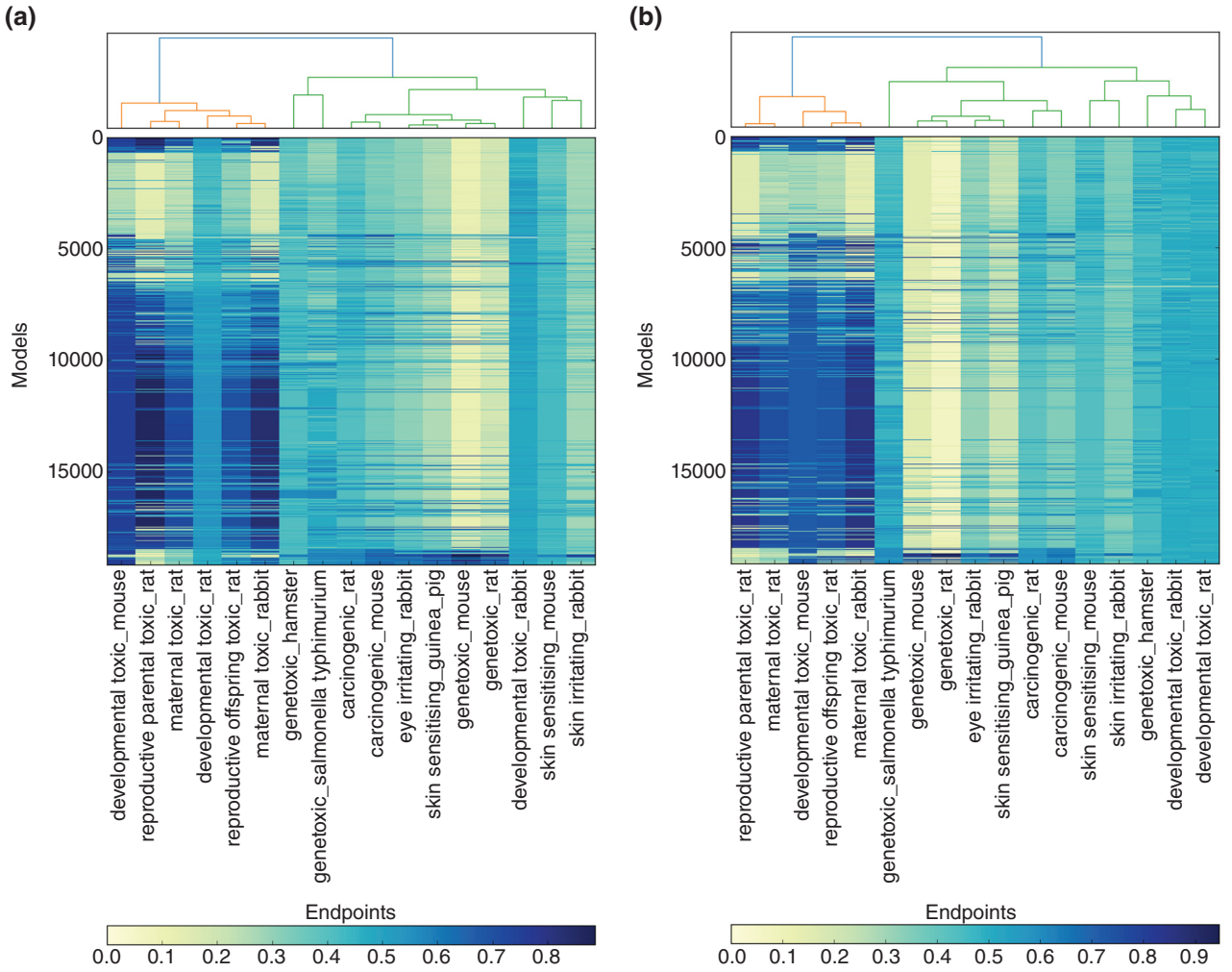

FIGURE 10 | Predictability of endpoints in (a) internal and (b) external validation. The heat maps show models' performances in predicting each toxicity endpoint. Each row corresponds to a model, and each column corresponds to a toxicity endpoint. Cells represent model's performance in predicting each endpoint. Models are numbered from 0 to 19,185. The performance is calculated using mean absolute error metric and ranges from 0.0 (best performance) to 1.0 (worst performance). The endpoints were clustered according to models' performances in predicting the endpoints into two clusters: endpoints with high predictability (green clusters) and endpoints with low predictability (orange clusters).

Figure 11 show models' performances using the MAE metric, calculated per compound, in internal (Figure 11(a)) and external (Figure 11(b)) validations. The compounds were clustered into three groups with high, medium, and low predictability. We used the Euclidean distance to calculate the distance between pairs of compounds. To calculate the distance between pairs of clusters, we used the average linkage method, which is the average distance between pairs of compounds from each cluster. We considered compounds to have high predictability if the majority of the models achieved high performance in predicting their toxicities; otherwise, the compounds have low predictability. Some compounds have medium predictability, which are the compounds that could not be clustered with the compounds with high or low predictability. The green, purple, and orange clusters in Figure 11 contain compounds with high, medium, and low predictability, respectively. Nevertheless, some models achieved high performance in predicting poorly predictable compounds.

This analysis indicates that it is more challenging to predict the toxicity of some compounds than the others. Therefore, we used Chi-square test[53] to identify chemical features that can distinguish between these three sets of compounds (compounds with high, medium, or low predictability). Figure S3 shows ranked features according to their Chi-square scores in internal (Figure S3(a)) and external

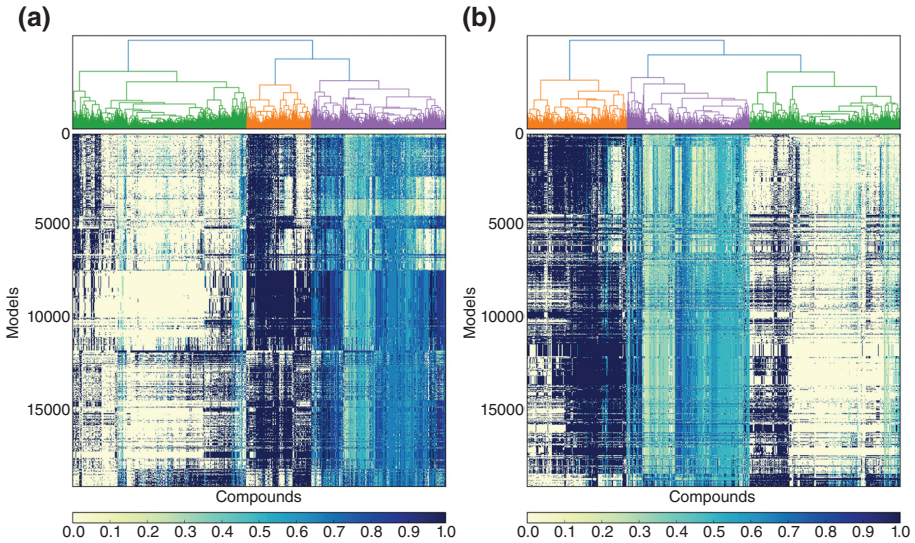

FIGURE 11 | Predictability of compounds' toxicity in (a) internal and (b) external validations. The heat maps show models performances in predicting the toxicity of each compound. Each row corresponds to a model, and each column corresponds to a compound. Cells represent each model's performance in predicting the toxicity of each compound. Models are numbered from 0 to 19,185. The performance is calculated using mean absolute error metric and ranges from 0.0 (best performance) to 1.0 (worst performance). The compounds were clustered into three groups according to models' performances in predicting the compounds toxicities: compounds with high predictability (green clusters), compounds with medium predictability (magenta clusters), and compounds with low predictability (orange clusters).

(Figure S3(b)) validations. Features with low ranks (the rightmost side in each bar graph) are present with similar frequencies in the three sets of compounds, whereas features with high ranks (leftmost side) are present with different frequencies in the three sets of compounds.

Moreover, we analyzed the relationship between compounds predictability and the number of known toxicity effects per compound. The histograms in Figure 12 show probability distribution of the number of toxicity effects for compounds with high, medium, and low predictability in internal (Figure 12(a)) and external (Figure 12(b)) validations. Notably, the number of known toxicity effects for compounds with high predictability ranges from 1 to 8, and 1 to 10 in internal and external validation, respectively. Similarly, the number of known toxicity effects for compounds with low predictability ranges from 1 to 8, and 1 to 6 in internal and external validation, respectively. However, compounds with medium predictability have 1–15, and 1–14 known toxicity effects per compound for internal and external validation, respectively. In the internal validation, the average number of known toxicity effects for compounds with high, medium and low predictability is 1.66, 4.47, and 1.33, respectively, while it is 1.81, 4.50, and 1.40, respectively, in the external validation. We observe that compounds that have high and low predictability are associated with a small number of known toxicity effects. However, compounds with medium predictability are associated with a larger number of known toxicity effects.

Effect of Feature Selection on Model Performance

Figure S4 illustrates the relationship between the number of selected features and models performance for five metrics in internal (Figure S4(a)) and external (Figure S4(b)) validations. While there is no association between the number of selected features and models' performances, some models achieve high performance using only a small subset of features. Moreover, Figure 13 shows that no feature selection method strictly outperformed others, and some models achieved good performance even when no feature selection method is applied.

CONCLUSION AND OUTLOOK

We realize that our conclusions are constrained by the data set and methods that we considered in this study. However, we believe that the diversity of the compounds and computational methods are sufficient to draw meaningful conclusions. This study illustrates the advantages of using multi-label models for toxicity assessment of pharmaceutically, environmentally, and industrially important compounds, even when toxicity data are partially available. The results of this comprehensive analysis of the state-of-the-art

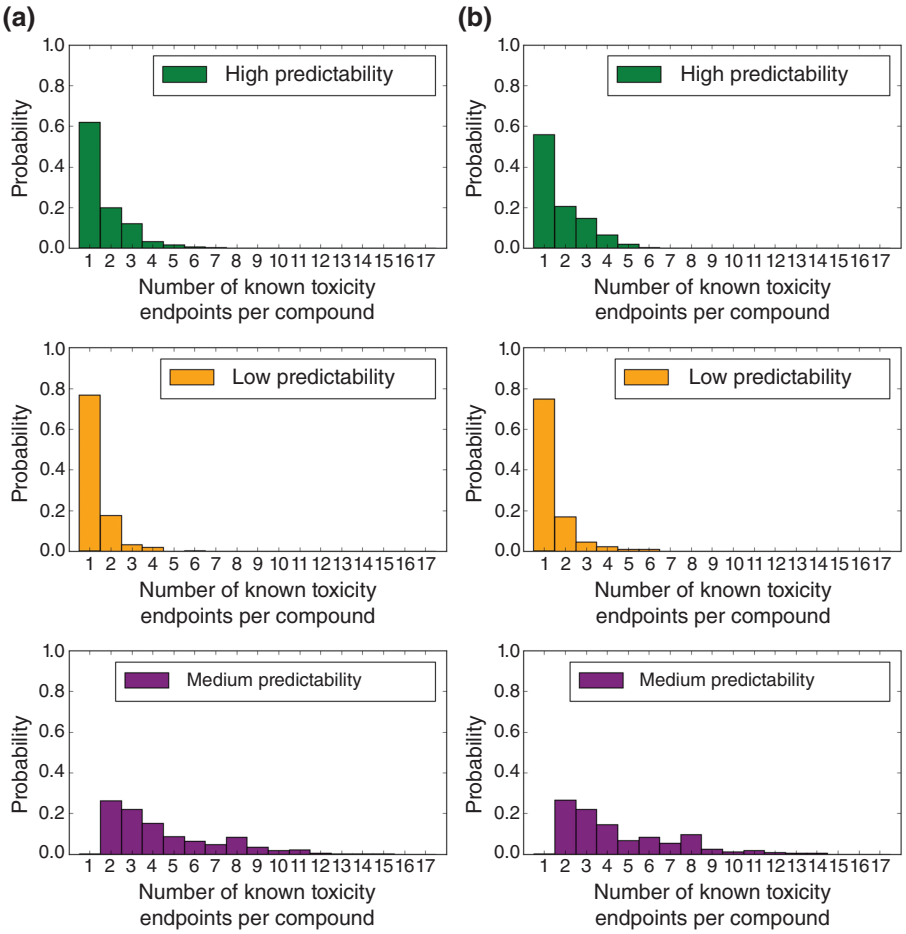

FIGURE 12 | The relationship between compounds predictability and the number of known toxicity effects per compound. The histograms show the probability distribution of the number of known toxicity endpoints per compound for compounds with high, medium, and low predictability in (a) internal and (b) external validations.

methods for multi-label classification demonstrates these models' abilities to exceed the performance of binary relevance models, suggesting that combining endpoints correlations with compound's features can increase model performance. Moreover, multi-label methods can reduce the number of generated models. When using the binary relevance method, one model must be created per endpoint (i.e., 17 models for our data set). However, the best-performing multi-label classification method in this benchmark study required generating only five models to predict all of the 17 endpoints.

We observed variability in models' predictive performances across endpoints. A primary factor influencing their performance could be the ratio of toxic and nontoxic compounds affiliated with each endpoint. Figure 4(b) shows five endpoints associated with a large number of toxic compounds, but a small number of nontoxic compounds (i.e., unbalanced endpoints) namely: reproductive paternal toxicity in rats, reproductive offspring toxicity in rats, maternal toxicity in rats, maternal toxicity in rabbits, and developmental toxicity in mice. Notably, Figure 10 shows that these five endpoints have low predictability. However, this observation does not hold in the opposite situation when the number of nontoxic compounds is much larger than the number of toxic compounds. Two endpoints fall in this category as shown in Figure 4(b): genotoxicity in mice and genotoxicity in rats. Figure 10 demonstrates that these two endpoints have high predictability.

Moreover, we recognized that the number of the compounds associated with each endpoint did not influence models' performances based on the data set we used. Figure 4(b) shows that four toxicity endpoints namely: developmental toxicity in mice and rabbits, genotoxicity in rats and maternal toxicity in rabbits, are associated with a small number compounds (i.e., less than 500 compounds). However, Figure 10 shows that two of these endpoints

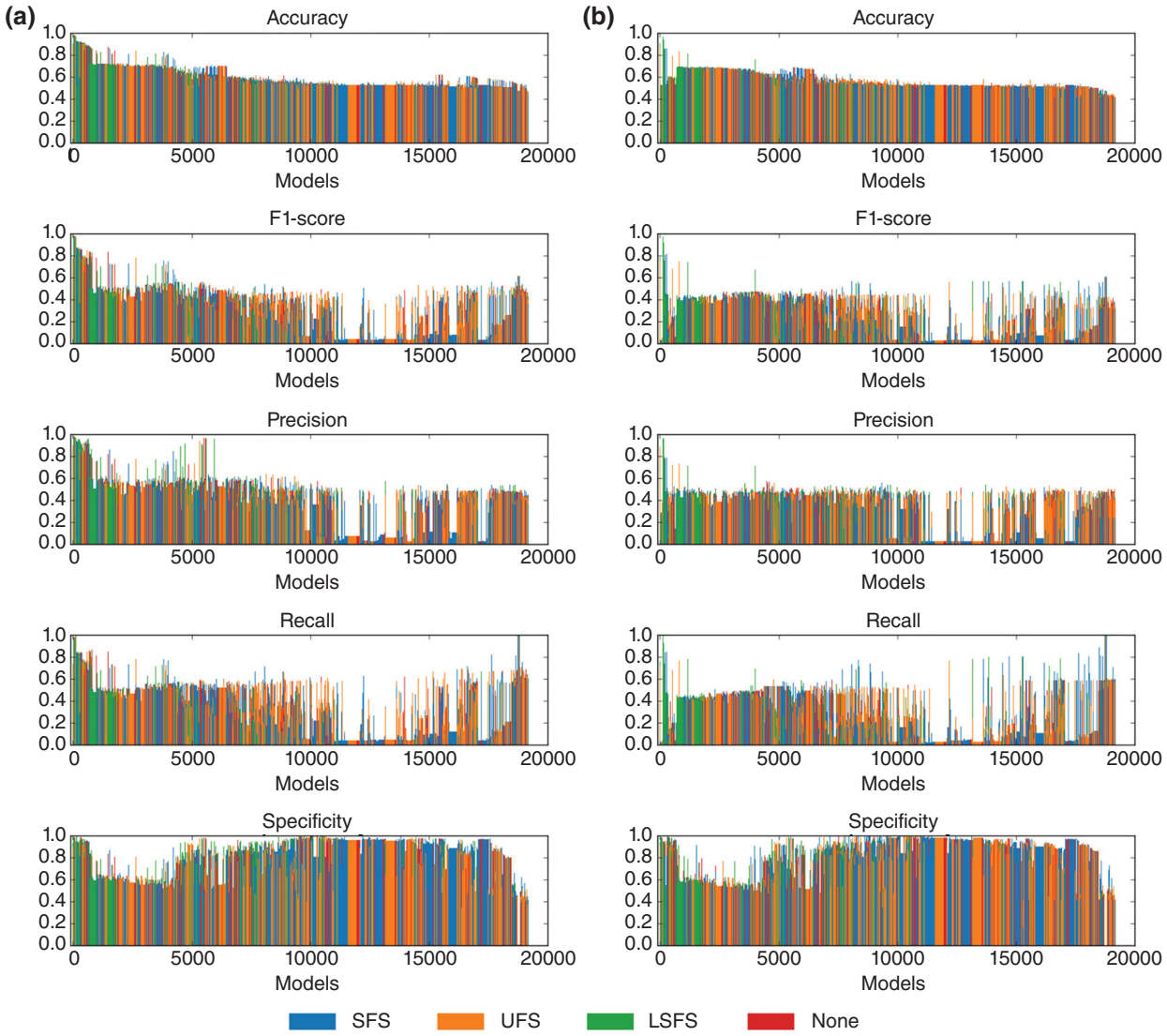

FIGURE 13 | Effect of feature selection on models' performance in (a) internal and (b) external validation. Bar graphs show models' macro-average performance via five metrics: accuracy, F1-score, precision, recall, and specificity. Models are numbered from 0 to 19,185. SFS, supervised feature selection; UFS, unsupervised feature selection; LSFS, label-specific feature selection, None, no feature selection method is applied.

(developmental toxicity in mice and maternal toxicity in rabbits) have low predictability, but the remaining two endpoints (developmental toxicity in rabbits and genotoxicity in rats) have high predictability.

When examining the choices of methods for multi-label classification, base classifiers and feature selection with their performances in internal and external validations, we observe that no single method strictly outperformed others. Therefore, models performance is dependent on a strategic application of combinations of these methods. Additionally, some models exhibited similar performance, although they were generated by different combinations of the methods (Table S2). However, the fact that the top-performing models provided in Table 2 were generated using the Random *K* Labelset method suggests that this method is a good candidate for modeling multi-label toxicity data with missing labels.

The methods applied in this study will be useful in future settings to predict narrow toxicity endpoints. In this study, we used broad endpoints of toxicity phenotypes in different species. Narrow endpoints may include strain, gender, age, dose, tissues or organs, and route and duration of exposure. Figure S5 shows an example of a hierarchy of toxicity endpoints. The broad endpoints (also called composite endpoints[14]) are at the top of the hierarchy. However, the lower we go down the hierarchy, the narrower are the endpoints. Predicting narrow

endpoints aids in identifying compounds' toxicity across different conditions. However, there are several challenges in modeling narrow endpoints:

- Few compounds are associated with each narrow endpoint. This limitation may inhibit models ability to identify useful features or detect outliers. However, broad endpoints aggregate toxicity data of compounds for all endpoints at the bottom of the hierarchy. Therefore, it is plausible that predictive models can achieve better performance in predicting broad rather than narrow endpoints.
- The number of narrow endpoints is much larger than broad endpoints. Developing a model for each narrow endpoint is infeasible, which renders the binary relevance method impractical. Applying multi-label methods will be necessary to reduce the number of generated models.
- Compounds are not tested for all narrow endpoints resulting in more missing toxicity data for each compound. Subsequently, identifying correlations between narrow endpoints, which is an essential component of multi-label methods, may become more challenging.

However, these challenges can be overcome once more data becomes available.

Toxicity endpoints are often measured in continuous values rather than binary categories. Predicting continuous endpoints can eliminate the discretization step of toxicity data, which requires endpoint-specific olds.[23] However, it was shown in another study that predicting exact values of continuous endpoints is harder than predicting binary categories.[14] Some multi-label classification methods, such as Random *K* labelset and label powerset, are not suitable for continuous endpoints. Therefore, future work should be directed to developing multi-label methods that can process continuous toxicity data.

Although the best-performing multi-label model achieved high performance, it may be useful to utilize other well-performing models that were generated in this study. An ensemble consisting of well-performing models with minimum overlapping of their correct predictions may outperform any single model.[18] On the other hand, combining predictions of good and poor models may hinder the performance of good models. Nevertheless, robust testing should be performed to evaluate the quality of aggregated predictions.

ACKNOWLEDGMENTS

Research reported in this publication were supported by the King Abdullah University of Science and Technology (KAUST) (BAS/1/1606-01-01) and by the KAUST Office of Sponsored Research (OSR) under Awards No URF/1/1976-02.

FURTHER READING

Appice A, Ceci M, Loglisci C, Manco G, Masciari E, Ras Z, eds. *New Frontiers in Mining Complex Patterns*. 1st ed. Cham, Switzerland: Springer; 2014.

Pardalos PM, Boginski VL, Alkis V, eds. *Data Mining in Biomedicine*. 1st ed. Gainesville, FL: Springer; 2007.

Dijkstra TMH, Tsivtsivadze E, Marchiori E, Heskes T, eds. *Pattern Recognition in Bioinformatics*. 1st ed. Berlin and Heidelberg, Germany: Springer; 2010.

Wang JTL, Zaki MJ, Toivonen H, Shasha D, eds. *Data Mining in Bioinformatics*. 1st ed. London, UK: Springer-Verlag; 2005.

REFERENCES

1. Arome D, Chinedu E. The importance of toxicity testing. *J Pharm Bio Sci* 2013, 4:146–148.
2. Parasuraman S. Toxicological screening. *J Pharmacol Pharmacother* 2011, 2:74–79.
3. Auletta AE, Dearfield KL, Cimino MC. Mutagenicity test schemes and guidelines: U.S. EPA office of pollution prevention and toxics and office of pesticide programs. *Environ Mol Mutagen* 1993, 21:38–45.

4. Rulis AM, Hattan DG. FDA's priority-based assessment of food additives: II general toxicity parameters. *Regul Toxicol Pharmacol* 1985, 5:152–174.
5. Shukla SJ, Huang R, Austin CP, Xia M. The future of toxicity testing: a focus on in vitro methods using a quantitative high-throughput screening platform. *Drug Discov Today* 2010, 15:997–1007.
6. Raies AB, Bajic VB. In silico toxicology: computational methods for the prediction of chemical toxicity. *WIREs Comput Mol Sci* 2016, 6:147–172.
7. Bliss CI. The method of PROBITS. *Science* 1934, 79:38–39.
8. Matthews EJ, Kruhlak NL, Cimino MC, Benz RD, Contrera JF. An analysis of genetic toxicity, reproductive and developmental toxicity, and carcinogenicity data: II. Identification of genotoxicants, reprotoxicants, and carcinogens using in silico methods. *Regul Toxicol Pharmacol* 2006, 44:97–110.
9. Zhang M-L, Zhou Z-H. A review on multi-label learning algorithms. *IEEE Trans Knowl Data Eng* 2014, 26:1819–1837.
10. Lhasa Limited. Derek Nexus. 2017. Available at: https://www.lhasalimited.org/products/derek-nexus.htm. (Accessed September 9, 2017).
11. Patlewicz G, Jeliazkova N, Safford R, Worth A, Aleksiev B. An evaluation of the implementation of the Cramer classification scheme in the Toxtree software. *SAR QSAR Environ Res* 2008, 19:495–524.
12. Ideaconsult Ltd. ToxTree. 2015. Available at: https://eurl-ecvam.jrc.ec.europa.eu/laboratories-research/predictive_toxicology/qsar_tools/toxtree. (Accessed September 9, 2017).
13. OECD. The OECD QSAR Toolbox. 2016. Available at: http://www.oecd.org/chemicalsafety/risk-assessment/oecd-qsar-toolbox.htm. (Accessed August 15, 2016).
14. Matthews EJ, Kruhlak NL, Cimino MC, Benz RD, Contrera JF. An analysis of genetic toxicity, reproductive and developmental toxicity, and carcinogenicity data: I. Identification of carcinogens using surrogate endpoints. *Regul Toxicol Pharmacol* 2006, 44:83–96.
15. Mishra M, Fei H, Huan J. Computational prediction of toxicity. *Int J Data Min Bioinform* 2013, 8:338–348.
16. Jeliazkova N, Jeliazkov V. Hierarchical multi-label classification of ToxCast datasets. In: *ToxCast Data Analysis Summit*. US EPA, Research Triangle Park, NC; 2009.
17. Mayr A, Klambauer G, Unterthiner T, Hochreiter S. DeepTox: toxicity prediction using deep learning. *Front Environ Sci* 2016, 3.
18. Eduati F, Mangravite LM, Wang T, Tang H, Bare JC, Huang R, Norman T, Kellen M, Menden MP, Yang J. Prediction of human population responses to toxic compounds by a collaborative competition. *Nat Biotechnol* 2015, 33:933–940.
19. Huang R, Xia M, Sakamuru S, Zhao J, Shahane SA, Attene-Ramos M, Zhao T, Austin CP, Simeonov A. Modelling the Tox21 10 K chemical profiles for in vivo toxicity prediction and mechanism characterization. *Nat Commun* 2016, 7:10425.
20. Jiang Z, Xu R, Dong C. Identification of chemical toxicity using ontology information of chemicals. *Comput Math Methods Med* 2015, 2015:246374.
21. Chen L, Lu J, Zhang J, Feng K-R, Zheng M-Y, Cai Y-D. Predicting chemical toxicity effects based on chemical-chemical interactions. *PLoS One* 2013, 8: e56517.
22. Batke M, Bitsch A, Gundert-Remy U, Gütlein M, Helma C, Kramer S, Maunz A, Partosch F, Seeland M, Stahlmann R. Multi-label-classification to predict repeated dose toxicity in the context of REACH. *Naunyn Schmiedebergs Arch Pharmacol* 2014, 387:S45.
23. Batke M, Gütlein M, Partosch F, Gundert-Remy U, Helma C, Kramer S, Maunz A, Seeland M, Bitsch A. Innovative strategies to develop chemical categories using a combination of structural and toxicological properties. *Front Pharmacol* 2016, 7:321.
24. Schafer JL. *Analysis of Incomplete Multivariate Data*. Boca Raton, FL: CRC press; 1997.
25. Montanari F, Zdrazil B, Digles D, Ecker GF. Selectivity profiling of BCRP versus P-gp inhibition: from automated collection of polypharmacology data to multi-label learning. *J Chem* 2016, 8:7.
26. Kotsiantis SB. Supervised machine learning: a review of classification techniques. *Inf Dent* 2007, 31: 249–268.
27. Tsoumakas G, Katakis I. Multi-label classification: an overview. *Int J Data Wareh Min* 2007, 3:1–13.
28. Luaces O, Díez J, Barranquero J, Coz JJ, Bahamonde A. Binary relevance efficacy for multilabel classification. *Prog Artif Intell* 2012, 1:303–313.
29. Read J, Pfahringer B, Holmes G, Frank E. Classifier chains for multi-label classification. *Mach Learn* 2011, 85:333–359.
30. Tsoumakas G, Katakis I, Vlahavas I. Random k-labelsets for multilabel classification. *IEEE Trans Knowl Data Eng* 2011, 23:1079–1089.
31. Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. In: Bartlett P, Mansour Y, eds. *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*. Madison, WI: ACM; 1998, 92–100.
32. Wicker J, Pfahringer B, Kramer S. Multi-label classification using boolean matrix decomposition. In: *Proceedings of the 27th Annual ACM Symposium on Applied Computing 2012*, 2012, 179–186.
33. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* 2015, 521:436–444.
34. Clare A, King RD. Knowledge discovery in multi-label phenotype data. In: De Raedt L, Siebes A, eds.

Principles of Data Mining and Knowledge Discovery. Berlin and Heidelberg, Germany: Springer; 2001, 42–53.

35. Chiang T-H, Lo H-Y, Lin S-D. A ranking-based KNN approach for multi-label classification. In: Hoi SCH, Buntine W, eds. *Proceedings of the Asian Conference on Machine Learning*. PMLR: Singapore; 2012, 81–96.
36. Cover T, Hart P. Nearest neighbor pattern classification. *IEEE Trans Inf Theory* 1967, 13:21–27.
37. Podani J. Distance, similarity. In: *Introduction to the Exploration of Multivariate Biological Data*. Leiden, The Netherlands: Backhuys Publishers; 2000, 55–110.
38. Cox DR. The regression analysis of binary sequences. *J R Stat Soc Series B Stat Methodol* 1958, 20:215–242.
39. Ypma TJ. Historical development of the Newton–Raphson method. *SIAM Rev* 1995, 37:531–551.
40. Nocedal J. Updating quasi-Newton matrices with limited storage. *Math Comp* 1980, 35:773–782.
41. Wright SJ. Coordinate descent algorithms. *Math Prog* 2015, 151:3–34.
42. Schmidt M, Le Roux N, Bach F. Minimizing finite sums with the stochastic average gradient. *Math Prog* 2017, 162:83–112.
43. Ng AY. Feature selection, L 1 vs. L 2 regularization, and rotational invariance. In: Greiner R, Schuumans D, eds. *Proceedings of the twenty-First International Conference on Machine Learning*. Banff, Canada: ACM, 2004, 78–85.
44. Breiman L, Friedman JH, Olshen RA, Stone CJ. *Classification and Regression Trees*. Monterey, CA: CRC; 1984.
45. Breiman L. Random forests. *Mach Learn* 2001, 45:5–32.
46. Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. *Mach Learn* 2006, 63:3–42.
47. Boser BE, Guyon IM, Vapnik VN. A training algorithm for optimal margin classifiers. In: Haussler D, ed. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. Pittsburgh, PA: ACM; 1992, 144–152.
48. Burges CJ. A tutorial on support vector machines for pattern recognition. *Data Min Knowl Discov* 1998, 2:121–167.
49. Crammer K, Singer Y. On the algorithmic implementation of multiclass kernel-based vector machines. *J Mach Learn Res* 2002, 2:265–292.
50. Zhou D, Bousquet O, Lal TN, Weston J, Schölkopf B. Learning with local and global consistency. In: *NIPS'03 Proceedings of the 16th International Conference on Neural Information Processing Systems*. Whistler, Canada, 2003.
51. Maron ME, Kuhns JL. On relevance, probabilistic indexing and information retrieval. *J ACM* 1960, 7:216–244.
52. Podani J. Matrix Rearrangement. In: *Introduction to the Exploration of Multivariate Biological Data*. Leiden, The Netherlands: Backhuys Publishers; 2000, 285–311.
53. Powers DMW. Evaluation: from precision, recall and f-measure to ROC, informedness, markedness & correlation. *J Mach Learn Tech* 2011, 2:37.
54. Huang Q, Tao D, Li X, Jin L, Wei G. Exploiting local coherent patterns for unsupervised feature ranking. *IEEE Trans Syst Man Cybern B Cybern* 2011, 41:1471–1482.
55. He Y, Liew CY, Sharma N, Woo SK, Chau YT, Yap CW. PaDEL-DDPredictor: open-source software for PD-PK-T prediction. *J Comput Chem* 2013, 34:604–610.
56. Ellison CM, Sherhod R, Cronin MT, Enoch SJ, Madden JC, Judson PN. Assessment of methods to define the applicability domain of structural alert models. *J Chem Inf Model* 2011, 51:975–985.
57. Jaworska J, Nikolova-Jeliazkova N. How can structural similarity analysis help in category formation? *SAR QSAR Environ Res* 2007, 18:195–207.
58. Ideaconsult Ltd. Ambit discovery. 2006. Available at: http://ambit.sourceforge.net/download_ambitdiscovery.html. (Accessed September 19, 2017).
59. Dietterich T. Overfitting and undercomputing in machine learning. *ACM Comput Surv* 1995, 27:326–327.
60. Willmott CJ, Matsuura K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Clim Res* 2005, 30:79–82.
61. Podani J. Hierarchical clustering. In: *Introduction to the Exploration of Multivariate Biological Data*. Leiden, The Netherlands: Backhuys Publishers; 2000, 135–164.

